(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 612 171 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: 94101561.2

(51) Int. Cl.5: H04L 12/56, H04Q 11/04

(22) Date of filing: 02.02.94

(30) Priority: 15.02.93 JP 25280/93

(43) Date of publication of application:
24.08.94 Bulletin 94/34

(84) Designated Contracting States:
DE FR GB SE

(71) Applicant: MITSUBISHI DENKI KABUSHIKI KAISHA
2-3, Marunouchi 2-chome
Chiyoda-ku
Tokyo 100 (JP)

(72) Inventor: Yamanaka, Hideaki, c/o Mitsubishi Denki K.K.
Tsushin System Kenkyusho,
1-1, Ofuna 5-chome
Kamakura-shi, Kanagawa 247 (JP)
Inventor: Saito, Hirotaka, c/o Mitsubishi Denki K.K.
Tsushin System Kenkyusho,
1-1, Ofuna 5-chome
Kamakura-shi, Kanagawa 247 (JP)
Inventor: Tsuzuki, Munenori, c/o Mitsubishi Denki K.K.
Tsushin System Kenkyusho,
1-1, Ofuna 5-chome
Kamakura-shi, Kanagawa 247 (JP)
Inventor: Yamada, Hirotoshi, c/o Mitsubishi Denki K.K.
Tsushin System Kenkyusho,
1-1, Ofuna 5-chome
Kamakura-shi, Kanagawa 247 (JP)
Inventor: Kondoh, Harufusa, c/o Mitsubishi Denki K.K.
LSI Kenkyusho,
1 Mizuhara 4-chome
Itami-shi, Hyogo 664 (JP)
Inventor: Oshima, Kazuyoshi, c/o Mitsubishi Denki K.K.
Tsushin System Kenkyusho,
1-1, Ofuna 5-chome
Kamakura-shi, Kanagawa 247 (JP)
Inventor: Notani, Hiromi, c/o Mitsubishi Denki K.K.
LSI Kenkyusho,
1 Mizuhara 4-chome
Itami-shi, Hyogo 664 (JP)

(74) Representative: Pfenning, Meinig & Partner
Mozartstrasse 17
D-80336 München (DE)

(54) Data queueing apparatus and ATM cell switch based on shifting and searching.

(57) A data queueing apparatus is disclosed which can process data switching and queueing using a smaller capacity memory and yet which reduces the data discard and loss possibility.

Received limited-length data at the input line is written in shift memories, which can shift the data to the next stage regardless of the destination of the data. Destination indicating bits, which indicate the destination of the data, are associated with each of shift memories A search circuit detects an asserted bit among the destination indicating bits corresponding to the output lines The data is extracted by a selector Then the selector transmits the data to the desired output line, and the data stored in the previous stage of the shift memory is shifted.

The memory is used in common for a plurality of the output lines, so that a lower possibility of data discard and loss can be attained in a smaller capacity memory

Rank Xerox (UK) Business Services
(3. 10/3 09/3 3 4)

Background of the Invention

This invention relates to an address controller of a data switching apparatus and a data switching apparatus having a shared buffer for temporary storing, delaying, and switching a limited length of data, such as blocked-frame information of various kinds of multimedia, for example, voice, data, image, etc., and fixed-length packets.

Description of the Related Art

In ATM (Asynchronous Transfer Mode) communication systems, continuous signals such as line signals or voice signals, and bursty signals such as data signals or motion video are divided into fixed lengths of data, appended to a header indicating destination information, thus forming a packet. This packet is called a "cell" in ATM. Data are transferred in the same packet form. Synchronization of frames is not needed between terminal equipment and a channel, and the operating speed of the terminal equipment and the channel can be set independently. This system can be applied to many kinds of terminal equipment. But a high-speed packet switch receives packets at random, and a lot of packets may be addressed to one destination, so that packets need to be queued for processing to prevent loss of information.

To solve this problem, for example, a high-speed packet switch is proposed and shown in Figs. 5 and 6 of "PRELUDE: An Asynchronous Time-Division Switched Network," by Jean-Pierre Coudreuse and Michel Servel in International Conference on Communications 1987, session 22, document No.2. This document relates to a high-speed packet switch of an asynchronous transfer mode (ATM) communication system for efficiently multiplexing and transmitting line switching data or packet switching data. A conventional data queueing apparatus is shown in a controller 16 according to the above document. Fig. 30 shows a block diagram of an example of the conventional data queueing apparatus.

In the figure, n (n>=2) input ports 11_1 - 11_n are provided for incoming fixed-length packets. Also provided are m (m>=2) output ports 12_1 - 12_m for outgoing packets. Packet multiplexer 13 multiplexes received packets. Memory 14 is provided for writing data in a predefined address. Data can be read from the memory by indicating a read address regardless of the order of writing data. Packet demultiplexer 15 demultiplexes read-out packets. The controller 16 controls switching packets.

The controller 16 is shown in detail in Fig. 31. The controller 16 has a conventional data queueing apparatus shown in Fig. 10 of the above document. Fig. 31 shows the conventional data queueing apparatus with a little deformation for an explanation. In the figure, a header converter 17 chooses an address for the received cell to write in the memory 14. It also detects the destination output ports 12_1 - 12_m for the packet, and converts the received header into a new header. Cyclic selector 20 selects information sequentially from the conventional data queueing apparatus

The conventional data queueing apparatus 18 is in the controller 16. An input line 1 receives an address for writing a packet in the memory 14. Destination inputs 3_1 - 3_m receive destination information. Output lines 2_1 - 2_m are provided for transmitting addresses after queuing of addresses. FIFO memories 19_1 - 19_m are provided corresponding to each of output ports 12_1 - 12_m. The conventional data queueing apparatus 18 forms queues of the writing addresses of the received packets corresponding to the output ports 12_1 - 12_m, where the addresses queue up for the output ports 12_1 - 12_m in the received order.

The packets arriving at these input ports 11_1 - 11_n of the high-speed packet switch are multiplexed by the packet multiplexer 13 and written in the memory 14. A header having destination information is transmitted to the controller 16. The header converter 17 determines from the header the destination output ports 12_1 - 12_m and converts the header into a new header. Addresses of the packets in the memory 14 are lined up according to the output ports 12_1 - 12_m by the data queueing apparatus. The FIFO memories 19_1 - 19_m are used in the conventional data queueing apparatus 18.

The packets are read from the memory 14 according to the addresses, which are read from the conventional data queueing apparatus 18. The packets are demultiplexed by the packet demultiplexer 15 and launched to the predetermined output ports 12_1 - 12_m. As described above, the packets received at the input ports 11_1 - 11_n are transmitted to the desired output ports 12_1 - 12_m, thus the packets are switched.

Fig. 32 shows another example of a conventional ATM switch named shared multibuffer ATM switch according to the invention described in "622 Mb/s 8 x 8 Shared Multibuffer ATM Switch with Hierarchical Queueing and Multicast Functions", GLOBECOM '93, session #40, December 1993 by Hideaki Yamanaka et al. Another conventional data queueing apparatus is address queues in the controller 16. When a cell comes to one of the input ports 11_1 - 11_8, the ATM switch detects the header in the cell and transfers it to the controller 16. The incoming cell is written in one of the vacant addresses of the shared buffer memories (SBMs). This address is chosen in the controller 16. The controller 16 has plural address queues. Each address queue is for one of output ports.

Address queues are selected based on the destination of the cell. Then the write address is buffered in the selected address queues. On the other hand, the incoming cell is passed through a cross point switch 3201, transferred to a shared buffer memory (SBM), and stored there. The data stored in SBM are transmitted to one of the output ports 12_1 - 12_8 based on the output direction from the controller 16. Fig. 33 shows the address queues by the FIFO memories in the controller 16.

Fig. 34 shows an example of an operation for multicast cells, which are to be multicasted to a plurality of the output ports. For processing the multicast cells, multicast cell counters (MCCs) are provided. The MCCs correspond to the data stored in the shared buffer memory (SBM) and indicate if the received data are multicast cells or not. For example, when an incoming multicast cell is stored in SBM 14_1 in address A0 and this multicast cell is to be copied to two output ports, MCC 93 increments the counter value from "0" to "2" as shown in Fig. 34. Another cell stored in address B3 in SBM 14_2 is a multicast cell destined for two output ports, and MCC 90 decrements the counter value from "2" to "0" after the cell in B3 goes out to two output ports. As the MCC 90 becomes "0", the address B3 is released and put in idle address pool. Though another cell destined for two output ports is stored in SBM 14_3 as shown by MCC 92, it has been transferred to only one output port in this time slot. The MCC 92 decrements the counter value from "2" to "1". As the MCC 90 is not "0", the address C1 is not released. As a result, the cell in Cl can wait until the next slot for going out to the other destination port. In this way, the MCCs control to detect the release timing of addresses of SBM for multicast cells.

Fig 35 shows a graph of memory capacity required by the shared buffer memories (SBMs) and the address queues (AQs), where B is the size of the buffer memories per output port (cells/port). AQs are an example of a conventional data queueing apparatus. Fig. 36 shows a formula for drawing this graph. The required memory capacity of the address queues (AQ) increases according to the increase of the number of the output ports. The memory capacity of the address queues (AQ) is tuned to be larger than the memory capacity of SBM when the output ports number 64 or more.

Fig. 37 shows some considerations relative to the design of high-speed ATM switches and large-scaled ATM switches. The operation speed must be increased in the high-speed ATM switches. Therefore, the number of addresses written in the address queues per second is increased in proportion to the operation speed, the number of the input ports, and the number of the output ports. As a result, the address queues or a data queueing apparatus is forced to operate at a high speed. And the number of the input ports and the output ports must be increased in large-scaled ATM switches. The memory capacity of the shared buffer memory must be increased as well. In accordance with these increases, the number and the lengths of the queues need to be increased. The number of address bits also needs to be increased. Thus, the total memory capacity of the address queues gets larger.

In the conventional data queueing apparatus configured as described above, for example, when the memory 14 has capacity to hold P number of packets, the FIFO memory 19 needs to have capacity of P number of addresses so as to prevent a loss of packets caused by an overflow of addresses. Therefore, the data queueing apparatus 18 needs a memory capacity sufficient to hold P times as many as m number of addresses as a whole, so that the scale of the apparatus becomes large.

Summary of the Invention

The object of this invention is to solve the above problems. The invention provides an apparatus which does not need plural memories for storing data, but uses a common memory for storing data for all of the output ports. The capacity of the memory becomes smaller, so that the scale of the apparatus can also become smaller. In addition, the possibility of data loss can be reduced because of sharing effect.

A data queueing apparatus according to one aspect of the invention may include:

(a) a plurality of output lines for transmitting data;

(b) input means for receiving data with destination information which indicates at least one of the output lines;

(c) memory means, which is provided for the plurality of output lines, for storing data received at the input means; and

(d) search and select means, which searches outgoing data for each output lines among the data stored in the memory means with the destination information, and selects the searched data in the memory means for the corresponding output line for transmitting.

Variations of this aspect of the invention may include the following.

In the data queueing apparatus according to the invention, the memory means may include:

(a) a plurality of shift memories for shifting data and storing data therein;

(b) a plurality of destination memories each of which corresponds to both each shift memory and each output line and stores the destination information of data therein

In the data queueing apparatus according to the invention, the search and select means may include a plurality of search means each of which corresponds to each of the plurality of the output lines and searches destination information stored in the corresponding destination memories for finding data to transmit to the corresponding output line. In the data queueing apparatus according to the invention, the search and select means may further include a plurality of select means each of which corresponds to each of the plurality of output lines and selects data found by the search means and transmits data to the corresponding output lines.

The data queueing apparatus according to the invention may further include delete means for deleting data stored in the memory means after being selected and transmitted by the search and select means.

In the data queueing apparatus according to the invention, input means may receive data with destination information, indicative of destination output lines, and the delete means may delete data after they are transmitted to the plurality of output lines indicated by the destination information.

The data queueing apparatus according to the invention may further include a plurality of shift detectors, each of which corresponds to each of the shift memories, for requesting a previous shift memory to shift data when it detects selecting and transmitting of the data stored in the corresponding shift memory.

In the data queuing apparatus according to the invention, the input means may expect to receive data every time slot, which has predefined time length, the plurality of shift memories may shift data more than once every time slot, and the search and select means may search data once every predefined X (X>=2) time slot.

In the data queueing apparatus according to the invention, the input means may expect to receive data each time slot which has predefined time length, may include Y (Y>=2) number of input lines which are coupled to the Y number of shift memories respectively for receiving data once to the coupled shift memory for every time slot, and the plurality of shift memories may shift data at least Y number of times for every time slot.

In the data queueing apparatus according to the invention, the search and select means may include search means whose number is less than the number of output lines.

In the data queueing apparatus according to the invention, the search and select means may include select means whose number is less than the number of output lines

In the data queueing apparatus according to the invention, the input means may receive a serial signal and may include serial-parallel converter for transforming the serial signal to the parallel signal, and the memory means may receive the parallel signal from the serial-parallel converter.

In the data queueing apparatus according to the invention, the shift memory and the destination memory may be coupled together as a memory unit.

A data queueing system according to another aspect of the invention may include;

(a) first data queueing apparatus for processing data having priority;

(b) second data queueing apparatus for processing data having different priority;

(c) data distributing means for transferring data to one of the first and second data queueing apparatuses based on the priority of received data.

Variations on this aspect of the invention may include the following.

In the data queueing apparatus, the input means may receive the destination information along with priority for processing data, the memory means may store the destination information and the priority, and the search and select means may search the destination information for selecting data based on the stored priority

A data queueing apparatus according to yet another aspect of the invention may include:

(a) an input line for receiving a limited length of data defined by a predefined fixed time;

(b) a plurality of output lines for transmitting a limited length of data;

(c) destination input means, which corresponds to the input line, for receiving destination information which indicates one of the plurality of output lines corresponding to the data;

(d) a plurality of shift memories, which are connected serially, for storing received data and shifting stored data forward serially;

(e) a plurality of destination indicating bits, each of which corresponds to both of each of the shift memories and each of the output lines, and is asserted when the bit corresponds to a destination of data stored in one of the shift memories, for storing the destination received at the destination input means and shifting the stored destination synchronous with the shifting of the data stored in the corresponding shift memories;

(f) search circuits, each of which corresponds to each of the output lines, for searching backward for an asserted bit in the destination indicating bits; and

(g) selectors for selecting data from the plurality of shift memories to the output lines based on search results of the search circuits.

The data queueing apparatus according to the invention wherein one of a limited length of data is transmitted to the plurality of output lines, may further include shift detectors, each of which corresponds to each of the shift memories, for detecting that all of the destination indicating bits which correspond to one of the shift memories are not asserted and directing next shift memory and next destination indicating bits to shift data and the destination forward.

A data queueing method according to the invention may include the steps of:

(a) receiving data with destination information from an input line every time slot;

(b) storing data with the destination information in a common memory in an order;

(c) searching the destination information in the order for destinations corresponding to each of a plurality of output lines;

(d) selecting data for each of the plurality of output lines based on a search result;

(e) transmitting selected data to each of output lines.

In the data queueing method according to the invention, the receiving step may include the steps of:

(a) checking if the common memory is full of data; and

(b) discarding data received if the common memory is full of data.

The data queueing method according to the invention may further include the step of marking as deleted that data transmitted from the memory after the transmitting step.

In the data queueing method according to the invention, the storing step may include the steps of:

(a) checking if the common memory holds data marked as deleted;

(b) shifting other data in the common memory onto the data marked as deleted if the common memory holds the data marked as deleted;

In the data queueing method according to the invention, the searching step may include the step of repeating the receiving step and the storing step a plurality of times before searching the destination information

An Asynchronous Transfer Mode switch according to yet another aspect of the invention, for switching cells, having destination information therewith, between input ports and output ports, may include:

(a) a shared multibuffer having addresses for storing cells;

(b) a controller having a data queueing apparatus providing the addresses, whereat the cells are stored in the shared multibuffer with the destinations of the stored cells to the data queueing apparatus, whereby the data queueing apparatus transmits the addresses of cells to be transmitted to the output ports

An Asynchronous Transfer Mode switch according to yet another aspect of the invention for switching cells, having destination information therewith, between input ports and output ports, may include:

(a) a shared multibuffer having addresses for storing cells;

(b) a controller having a common address queue for queueing the addresses of cells stored in the shared multibuffer and the destination information

In the Asynchronous Transfer Mode switch according to this invention, the address queue may include memory means for storing the addresses and the destination information of cells, and select means for selecting addresses for each of output ports based on the destination information.

In the Asynchronous Transfer Mode switch, the memory means may include shift memories for storing addresses and destination information, and searching elements for searching destination information stored in the shift memories.

In the Asynchronous Transfer Mode switch, each of the shift memories may include an occupation indicator for indicating an occupation status of the shift memory, and a shift trigger for shifting addresses and destination information to another shift memory based on the occupation status indicated by the occupation indicator in the other shift memory

Brief Description of the Drawings

In the drawings, Figure 1 is a block diagram of the data queueing apparatus according to an embodiment of this invention;

Figure 2 is a timing chart which shows timing of the operation of each part shown in the block diagram of Fig. 1;

Figure 3 is a timing chart which shows timing of the operation of each part shown in the block diagram of Fig. 1;

Figure 4 is a timing chart which shows timing of the operation of each part shown in the block diagram of Fig. 1;

Figure 5 is a flow chart of the operation according to the embodiment of this invention;

Figure 6 is a block diagram of the data queueing apparatus according to another embodiment of this invention;

Figure 7 is a timing chart which shows timing of the operation of each part shown in the block diagram of Fig. 6;

Figure 8 is a timing chart which shows timing of the operation of each part shown in the block diagram of Fig 6;

Figure 9 is a timing chart which shows timing of the operation of each part shown in the block diagram of Fig 6;

Figure 10 is a part of a block diagram of the data queuing apparatus according to another embodiment of this invention;

Figure 11 is a block diagram of the data queuing apparatus according to another embodiment of this invention;

Figure 12 is a block diagram of a configuration having a plurality of the data queueing apparatuses according to another embodiment of this invention;

Figure 13 is a block diagram of the data queueing apparatus controlling the priority according to another embodiment of this invention;

Figure 14 is a priority table for controlling the priority according to the embodiment of this invention;

Figure 15 is a count table for controlling the priority according to the embodiment of this invention;

Figure 16 is a block diagram of the serial-parallel converter according to another embodiment of this invention;

Figure 17 is a block diagram of the memory configuration of the data queueing apparatus according to the other embodiment of this invention;

Figure 18 is a block diagram of the apparatus comprising smaller numbers of selectors according to another embodiment of this invention;

Figure 19 shows the operation of the apparatus comprising smaller numbers of the selectors according to another embodiment of this invention;

Figure 20 is a block diagram of the apparatus comprising one detector and one selector according to another embodiment of this invention;

Figure 21 shows the operation of the apparatus comprising one detector and one selector according to another embodiment of this invention;

Figure 22 is a block diagram showing the shared multibuffer ATM switch according to another embodiment of this invention;

Figure 23 show a configuration of the data queueing apparatus in the shared multibuffer ATM switch according to the embodiment of this invention;

Figure 24 shows the shift operation according to the embodiment of this invention;

Figure 25 shows the operation of forward shifting according to the embodiment of this invention;

Figure 26 is a chart comparing the memory size of the data queueing apparatus according to this invention and the conventional data queueing apparatus;

Figure 27 shows a formula for calculating the memory size of the data queueing apparatus according to this invention and the conventional data queueing apparatus;

Figure 28 shows the shift operation according to another embodiment of this invention;

Figure 29 shows the operation of the searching element according to another embodiment of this invention;

Figure 30 is a block diagram of the high speed packet switch including a conventional data queueing apparatus;

Figure 31 is a block diagram of the conventional data queueing apparatus;

Figure 32 shows a configuration of the shared multibuffer ATM switch;

Figure 33 shows the address queueing by the conventional FIFO memories;

Figure 34 shows the operation of the conventional data queueing apparatus, when receiving a conventional multicast cell;

Figure 35 shows the memory size needed by the conventional shared buffer memories (SBMs) and the address queues (AQs);

Figure 36 shows a formula for calculating the memory size needed by the conventional shared buffer memories (SBMs) and the address queues (AQs); and

Figure 37 shows design considerations related to high-speed ATM switches and large-scaled ATM switches.

Description of the Preferred Embodiments

Embodiment 1.

The following is an explanation of a first embodiment, to be read while referring to the drawings. Fig. 1 shows an embodiment of a data queueing apparatus according to the invention In the figure, the same reference designations as used in Fig. 31 are used for the same or similar parts.

In Fig. 1, an input line 1 receives a limited length of data and a plurality of output lines 2_1 - 2_m transmit data. Destination inputs 3_1 - 3_m are provided correspondingly to the output lines 2_1 - 2_m and indicate destination output ports of received data. Shift memories 4_1 - 4_k store data Destination indicating bits 5_{11} - 5_{km} are provided correspondingly to each shift memory. The destination indicating bit 5_{12}, for example, shows whether the destination of the data stored in the shift memory 4_1 is the output line 2_2 or not The destination indicating bit 5_{km}, for another example, shows whether the destination of the data stored in the shift memory 4_k is the output line 2_m or not. Search circuits 6_1 - 6_m are provided correspondingly to each output lines 2_1 - 2_m. The search circuit 6_1, for example, is connected to the destination indicating bits 5_{11}, 5_{21} .. 5_{k1}. Selectors 7_1 - 7_m are provided correspondingly to each of the output lines 2_1 - 2_m. The selector 7_1, for example, corresponding to the output line 2_1, selects the data to be transmitted from the shift memories 4_1 - 4_k based on a search result of the search circuit 6_1, and transmits the data selected to the output line 2_1.

Operation of the apparatus of Fig. 1 is explained hereinafter. Figs. 2 - 4 show changing signals and memory contents when the number of the input line 1 is one, the number m of the output lines 2_1 - 2_m is four. the number k of the shift memories 4_1 - 4_k is six.

In Figs 2 - 4, line (a) shows time slots divided by a fixed length of time. Line (b) shows an example of data received in the input line 1, and line (c) shows destination inputs 3_4, 3_3, 3_2, and 3_1 in this order. Examples of the contents of the shift memories 4_1 - 4_6 and the destination indicating bits 5_{11} - 5_{64} in each time slot are shown in lines (d) - (o). Lines (p) - (s) show examples of the data to be transmitted to the output lines 2_1 - 2_4.

Data received in the input line 1 is a limited length of information such as frame information divided into predetermined fixed times, fixed-length address information, or fixed-length packet information In Figs. 2 - 4, the predetermined fixed times are the time slots and a serial number is assigned to each time slot from time slot 1 for the convenience of this explanation In these figures, one data item is received in each time slot.

These figures show an example in which no data was received before the time slot 1 or a long time has passed since data was last received, and therefore the shift memories 4_1 - 4_6 are empty before time slot 1. One data item is received in each of the time slots 1, 2, 3, 4, 6, 7, and 8.

In Figs. 2 - 4, the data to be transmitted to the output lines 2_1 - 2_4 are read once every 4 time slots. Actually, data is read in the time slot 4 and the time slot 8.

Destination information is received at the destination inputs 3_1 - 3_4 simultaneously with data received at the input line 1 For example, when the destination of the received data is the output line 2_4, only the destination input 3_4 is asserted. In the figures, "1" means an asserted bit, thus, in this case, destination inputs {3_4, 3_3, 3_2, 3_1} become {1, 0, 0, 0}.

"Data a" addressed to the output line 2_4 is received in the time slot 1, and "data b" addressed to the output line 2_1 is received in the time slot 2. In the time slot 1, the shift memory 4_6 is empty, thus, "data a" is shifted to the shift memory 4_6 immediately Destination inputs 3_1 - 3_4 are stored in the destination indicating bits 5_{64} -5_{61}

In the same way, in the time slot 2, the shift memory 4_5 is empty, thus, "data a" is shifted to the shift memory 4_5, and "data b" is shifted to the shift memory 4_6. When "data a" is shifted from the shift memory 4_6 to the shift memory 4_5 the destination in the destination indicating bits 5_{64} - 5_{61} are shifted to the destination indicating bits 5_{54} - 5_{51} corresponding to the next shift memory 4_5.

The following is an explanation of reading out data to the output lines 2_1 - 2_4. Figure 2 shows an example of transmitting data in the time slot 4. At the first stage of the time slot 4, the search circuits 6_1 - 6_4 search for and detect data to transmit. When any one of destination indicating bits 5_{11} - 5_{64} is asserted, a search result to that effect is passed to the selectors 7_1 - 7_4. The selectors 7_1 - 7_4 select one transmitted data of those detected for each output line 2_1 - 2_4 from the shift memories 4_1 - 4_6, and transmit data to the output lines 2_1 - 2_4

For example, the search circuit 6_1 detects data to transmit to the output line 2_1. The destination indicating bits 5_{11}, 5_{21}. 5_{31}, 5_{41}, 5_{51}, 5_{61} are read in this order until the asserted bit "1" is detected among these bits. If the asserted bit is detected, it is passed to the selector 7_1 The selector 7_1 shown in these figures selects one out of six. The selector 7_1 selects one of 6 shift memories 4_1 - 4_6, and the selected data is transmitted to the output line 2_1.

The above operation of transmitting data to the output line 2_1 is done independently of transmitting data to the other output lines 2_2 - 2_4. These operations of transmitting data to the output lines can be done separately and simultaneously.

The destination indicating bits 5_{11} - 5_{64} detected by the search circuits 6_1 - 6_4 become deasserted after selection based on a detection. The data stored in the shift memories 4_1 - 4_6 are deleted after reading. The next data is shifted to the shift memories 4_1 - 4_6, if there is any data in the previous stage.

The following is a more detailed explanation of the operation, referring to Figs. 2 - 4.

In each time slot 1 - 4, each of "data a, b, c, and d" is received. Each of "data a, b, c, and d" is stored in the shift memories in the order received.

In the time slot 4, one data is read and transmitted to each of the output lines 2_1 - 2_4 as described above.

The data to be transmitted to the output line 2_1 is explained referring to Fig. 2 hereinafter In the time slot 4, the search circuit 6_1 reads out the destination indicating bits 5_{11}, 5_{21}, 5_{31}, 5_{41}, 5_{51}, and 5_{61} in this order. These bits are 0, 0, 0, 0, 1, and 1 The destination indicating bit 5_{51} is detected from these values, because it is asserted and read before 5_{61}, and it is passed to the selector 7_1. The selector 7_1 selects the shift memory 4_5 from six shift memories 4_1 - 4_6 based on the result of the search circuit 6_1 operation, and "data b" is transmitted to the output line 2_1.

When "data b" is transmitted to the output line 2_1, the shift memory 4_5, in which "data b" was stored in the time slot 4, becomes empty. In the time slot 5, "data c", which was stored in the previous stage in the time slot 4, is shifted to the shift memory 4_5 as shown in Fig. 3. And the destination information in the destination indicating bits 5_{54}, 5_{63}, 5_{62}, and 5_{61} are also shifted to the destination indicating bits 5_{54}, 5_{53}, 5_{52}, and 5_{51}

In the same way, "data a" is transmitted to the output line 2_4.

Each value of the destination indicating bits 5_{12}, 5_{22}, 5_{32}, 5_{42} 5_{52}, and 5_{62} which correspond to the output line 2_2, is 0, 0, 0, 0, 0, and 0. This means there exists no asserted bits, thus the search circuit 6_2 informs the selector that there is no data to be transmitted. Thus, no data is transmitted from the output line 2_2.

In the same way, no data is transmitted from the output line 2_3.

No data is received in the time slot 5, however the data stored in the shift memories are shifted to the next shift memories. In the time slots 6 - 8, "data e, f, and g" are received sequentially. The received data are stored in the shift memories, and then shifted to the next shift memories.

In the time slot 8, data are read to be transmitted to the output lines as described above. In this case, "data c" is detected as the data to be transmitted to the output line 2_1 by the search circuit 6_1. "Data e" is detected as the data to be transmitted to the output line 2_2 by the search circuit 6_2 These detected "data c and e" are transmitted to each output lines The search circuits 6_3 and 6_4 can know there are no data to be transmitted to the output lines 2_3 and 2_4 because the destination indicating bits have no asserted bits. Accordingly, no data is transmitted to the output lines 2_3 and 2_4.

In the time slots 9 - 11, no data is received at the input lines. In these cases, the data stored in the shift memories are shifted to the next shift memories sequentially

"Data d" is stored in the shift memory 4_1 in the time slot 10 as shown in Fig 4. "Data d" cannot be shifted furthermore because the shift memory 4_1 is the last stage. "Data d" remains in the shift memory 4_1 during the time slot 11.

This "data d" is detected as the data to be transmitted to the output line 2_1 by the search circuit 6_1 in the time slot 12, then is transmitted to the output line 2_1 by the selector 7_1.

In the above examples, all received data could be written in the shift memory 4_6. But data cannot be written when all of the shift memories 4_1 - 4_6 are occupied by data. In this case, coming data are discarded. To reduce the possibility of data discard or loss, the number k of the shift memories 4_1 - 4_k is desired to be large.

A flow of the above operation is explained hereinafter referring to Fig 5.

At step S1, data and its destination are received at the input line 1 and the destination inputs 3_1 - 3_m.

At step S2, it is checked whether the first shift memory, which initially stores the received data, is occupied by any data. When the first shift memory has been occupied by some other data, the data and the destination received at step S1, are disposed at step S3. If it is detected that the first shift memory is empty at step S2, the first shift memory stores at step S5, the data and the destination which are received at step S1.

At step S6, it is checked if it is fourth input or not When it is not fourth input, the operation from step S1 - S5 is repeated. While the operation from step S1 - S6 is repeated, the received data are shifted at step S6', to the next shift memory one by one. By forward shifting as described above, the first shift memory, which receives the data for the first time, remains empty unless all shift memories are full.

If it is judged to be fourth input at step S6, the search circuits 6_1 - 6_4 search and detect the data to be transmitted to each output lines by checking the destination to each output lines at step S7.

At step S8, it is checked if the data to be transmitted to the output line is detected or not. When no data is detected to be transmitted, the operation is repeated from the step S1 again. When any data is detected to be transmitted at step S8, the data is selected from the shift memory by the selector at step S9.

At step S10, the data selected by the selector is transmitted to the corresponding output line.

At step S11, the data is deleted from the shift memory by clearing the corresponding destination since the data transmitted at step S10 is no longer needed. After deleting the data, the operation is repeated from receiving data and the destination at step S1. And the data stored in the previous shift memory is requested to shift to the next stage by this data deletion Thus, the data stored in the previous shift memory is overwritten in the shift memory where the destination has been cleared.- (Step S6')

As described above, in this embodiment, received limited-length data in the input line 1 are written in the shift memories, in which data can be shifted from one to the next sequentially regardless of its destination The destination indicating bits are associated with each of the shift memories. The search circuits search the data to be transmitted in received order by detecting an asserted bit from the destination indicating bits corresponding to the destination output lines, and the data is extracted by the selectors After extraction, the selector transfers the data to the desired output lines and the previous data is shifted to overwrite the transmitted data.

By this embodiment, the data received at the input line is written in the shift memories and its destination is stored in the destination indicating bits. Data are read in arrival order by searching the destination indicating bits corresponding to each of the output lines, so that the received data can be transmitted to the desired output line in arrival order. The shift memories can be used commonly for all output lines, so that the data queueing apparatus can reduce the possibility of data loss, which would be caused by attempting to write data, especially receiving bursty data which exceed the capacity of the shift memories

Embodiment 2.

The following is an explanation of another embodiment of this invention referring to the drawings. Fig 6 shows another embodiment of the data queueing apparatus according to this invention. The same reference designations are used for the same or similar elements as in Embodiment 1 and are not explained here, for clarity sake.

In Fig 6, shift detectors 8_1 - 8_k are provided correspondingly to each shift memories 4_1 - 4_k. The shift detectors trigger shifting of data stored in the previous shift memories 4_1 - 4_k when they detect the case that none of m number of destination indicating bits are asserted For example, when the shift detector 8_2, provided correspondingly to the shift memory 4_2, detects that none of destination indicating bits 5_{21} 5_{22}..5_{2m} are asserted, it shifts data stored in the previous shift memory 4_3.

Operation of this embodiment is explained hereinafter. Figs 7 - 9 are charts which show each signal and changing status of each element In this case, the number of the input lines 1 is one, the number m of output lines 2_1 - 2_m is four, and the number k of shift memories 4_1 - 4_k is six.

In Figs 7 - 9, line (a) shows time slots, line (b) shows an example of data received at the input line 1, and line (c) shows each of the destination inputs 3_4, 3_3, 3_2, and 3_1 for indicating the destination of the received data in this order. Lines (d) - (u) show examples of the status of the shift memories 4_1 - 4_6 and the destination indicating bits 5_{11} - 5_{64} in each time slot. And lines (v) - (y) show the data to be transmitted to each of the output lines 2_1 - 2_4.

The data received at the input line 1 is limited-length information such as frame information divided by a predefined fixed time, fixed length address information, or fixed length packet. In Figs. 7 - 9, the fixed-length time is defined as a time slot As shown in line (a), a serial number is assigned sequentially from time slot 1 for purposes of this explanation. In these figures, one data item is received in one time slot.

These figures show an example in which no data was received before the time slot 1 or a long time has passed since the last data was received, and the shift memories 4_1 - 4_6 therefore store no data just prior to the time slot 1. One data item is received in each of the time slots 1 2, 3 4, 6, 7, 8, 9, and 11.

In Figs. 7 - 9, the data to be transmitted to the output lines 2_1 - 2_4 are read out once every 4 time slots Actually, data is read out in the time slot 4, in the time slot 8 and the time slot 12.

Destination information is received at the destination inputs 3_1 - 3_4 simultaneously with data received at the input line 1. For example, when the destination of the received "data h" is the output line 2_4, the destination input 3_4 becomes asserted. In the figures, "1" means an asserted bit, thus, in this case, the destination inputs {3_4, 3_3, 3_2, 3_1} become {1, 0, 0, 0}.

When the received data is destined for a plurality of output lines, that is, multicast data, a plurality of bits of the destination inputs 3_1 - 3_4 become asserted For example, "data a" received in the time slot 1 is addressed to the output lines 2_1 and 2_4, thus the destination inputs 3_1 and 3_4 become asserted. In other words, the destination inputs {3_4, 3_3, 3_2, 3_1} become {1, 0, 0, 1}.

In the time slot 1, "data a" addressed to the output lines 2_1 and 2_4 is received, and "data b" addressed to the output line 2_2 is received in the time slot 2 In the time slot 1, the shift memory 4_6 is empty. "Data a" is therefore shifted to the shift memory 4_6 immediately The destination inputs 3_1 - 3_4 are stored in the destination indicating bits 5_{64} - 5_{61}.

In the same way, in the time slot 2, the shift memory 4_5 is empty Therefore, "data a" is shifted to the shift memory 4_5, and "data b" is shifted to the shift memory 4_6. With shifting "data a" from the shift memory 4_6 to 4_5, the destination in the destination indicating bits 5_{64} - 5_{61} are simultaneously shifted to the destination indicating bits 5_{54} - 5_{51}, which are provided correspondingly to the next shift memory 4_5.

The shift detectors 8_1 - 8_6 are provided correspondingly to the shift memories 4_1 - 4_6 In the time slot 1, they detect the case that none of m number of the destination indication among the destination indicating bits 5_{11} - 5_{64} is asserted For example, in the figures, as a status value of the shift detector, "1" is defined as meaning that at least one of the destination indicating bits is asserted, and "0" is defined that there is no asserted bit among the corresponding destination indicating bits Given this definition, OR gates can be used for the shift detectors 8_1 - 8_6.

For example, all of the destination indicating bits 5_{61} - 5_{64} corresponding to the shift memory 4_6 are "0" in the time slot 1, thus the status of the shift detector 8_6 becomes "0". In the time slot 2, one of the destination indicating bits 5_{61} - 5_{64} corresponding to the shift memory 4_6 is "1", thus the status of the shift detector 8_6 becomes "1".

The following is an explanation of reading out data to be transmitted to the output lines 2_1 - 2_4.

Fig. 7 shows an example of transmitting data in the time slot 4. At a first stage of the time slot 4, the search circuits 6_1 - 6_4 search for and detect data to be transmitted When any one of the destination indicating bits 5_{11} - 5_{64} is asserted, it is communicated to the corresponding selectors 7_1 - 7_4. The selectors 7_1 - 7_4 select one from the shift memories 4_1 - 4_6, and transfer data to the output lines 2_1 - 2_4.

For example, the search circuit 6_1 detects data to be transmitted to the output line 2_1. The destination indicating bits 5_{11}, 5_{21}, 5_{31}, 5_{41}, 5_{51}, 5_{61} are read in this order until the asserted bit "1" is detected among these bits. If the asserted bit is detected, it is communicated to the selector 7_1. The selector 7_1 in the figure is a selector which selects one from six. The selector 7_1 selects one from six shift memories 4_1 - 4_6, and the selected data is transferred to the output line 2_1

An operation of transmitting data to the output line 2_1 is carried out independently of transmitting data to the other output lines 2_2 - 2_4. These operations can be done separately and simultaneously.

The destination indicating bits 5_{11} - 5_{64} detected by the search circuits 6_1 - 6_4 are updated and become deasserted after selection of the corresponding data for transmit based on a detection. Then, each of the shift detectors 8_1 - 8_6 detect the possibility of shifting data to each of the corresponding shift memories 4_1 - 4_6 based on the updated destination indicating bits 5_{11} - 5_{64}. When it is noticed to be possible to shift data and the previous shift memory is occupied by data, the data is shifted to the corresponding shift memory.

The data to be transmitted to the output line 2_1 is explained referring to the figures hereinafter. In the time slot 4, the search circuit 6_1 reads out the destination indicating bits 5_{11}, 5_{21}, 5_{31}, 5_{41}, 5_{51}, and 5_{61} in this order. Each value of these bits is 0, 0, 0, 1, 0, and 1. The asserted destination indicating bit 5_{41} is detected from these values, and it is communicated to the selector 7_1. The selector 7_1 selects the shift memory 4_4 from six shift memories 4_1 - 4_6 based on the detection by the search circuit 6_1, and "data a" is transmitted to the output line 2_1

After "data a" is transferred to the output line 2_1, the corresponding destination indicating bit 5_{41} is cleared or deasserted.

In the same way, the search circuit 6_4 detects "data a" stored in the shift memory 4_4 as the data to be transferred to the output line 2_4 in the time slot 4. After "data a" is transferred to the output line 2_4, the corresponding destination indicating bit 5_{44} is cleared or deasserted

In the time slot 4, the destination indicating bits {5_{44}, 5_{43}, 5_{42}, 5_{41}} corresponding to the shift memory 4_4, which stored "data a", change value from {1, 0 0, 1} to {0, 0, 0, 0}. The shift detector 8_4 also changes status value from "1" to "0" at the same time.

When the shift detector 8_4 changes the status value from "1" to "0", that means "data a" is transmitted to all of the desired destinations, that is, the output lines 2_1 and 2_4. Then, "data a" is deleted

In the same way, "data b" in the shift memory 4_5 is transmitted to the output line 2_2 in the time slot 4. "Data c" stored in the previous shift memory 4_6 is shifted to the shift memory 4_5 in the time slot 5. And each of the destination indicating bits 5_{64}, 5_{63}, 5_{62}, and 5_{61} is also shifted to each of the destination indicating bits 5_{54}, 5_{53}, 5_{52}, and 5_{51} with shifting "data c" simultaneously.

Each of the destination indicating bits 5_{13}, 5_{23}, 5_{33}, 5_{43}, 5_{53}, and 5_{63}, which correspond to the output line 2_3, is 0, 0, 0, 0, 0, and 0 This means there exists no asserted bits, thus the search circuit 6_2 communicates the selector that there is no data to transmit. No data is transmitted from the output line 2_3.

In the time slot 5, no data is received. "Data e, f, and g" are received in the time slots 6 - 8

In the time slot 8, the search circuit detects the data to transmit. In this case, "data c" is to be transmitted to the output line 2_1. "Data d" is to be transmitted to the output line 2_2. Though "data d" is multicast data addressed to the output lines 2_1 and 2_2, "data c" is already chosen to be transmitted to the output line 2_1. Thus, "data d" is transmitted only to the output line 2_2. "Data e" is multicast data addressed to all output lines. But each of "data c and d" is already detected as the data to be transmitted to each of the output lines 2_1 and 2_2 in this case. "Data e" is thus transmitted to the output lines 2_3 and 2_4.

As a result, all of the destination indicating bits of "data c" become "0" and the shift detector changes the status value from "1" to "0" On the other hand, some of the destination indicating bits of "data d" and "data e" are still asserted, so that the status value of the shift detectors remain "1".

In the time slot 9, "data h" is received and no data is received in the time slot 10.

"Data d", stored in the shift memory 4_1 in the time slot 10, cannot be shifted further because the shift memory 4_1 is the last stage, so that "data d" remains in the shift memory 4_1 in the time slot 11.

This "data d" is transmitted to the output line 2_1 in the time slot 12 After "data d" is transmitted to the output line 2_1, no destination indicating bits are asserted, so that the shift detector changes the status value from "1" to "0". Thus,. "data d" is deleted.

In the above examples, all received data can be written in the shift memory 4_5. But received data cannot be written when all of the shift memories 4_1 - 4_5 are occupied by data. In this case, coming data are discarded. To reduce the possibility of data discard or loss, the number k of the shift memories 4_1 - 4_k is desired to be large.

As described above, in this embodiment, if multicast data addressed to a plurality of the output lines is received, the data can be shifted from one shift memory to the next, in accordance with whether the shift detectors detect that the data is transmitted to all desired output lines indicated by the destination indicating bits. The received data can be transmitted to the desired output line in the received order. The shift memories are each common to all output lines, so that the data queueing apparatus can reduce the possibility of data loss, which would be caused by attempting to write data, especially receiving bursty data which exceed the capacity of the shift memories

In this embodiment, the multicast data can be stored in the shift memories until the data is transmitted to all desired output lines, so that multicast processing can be done even if the shift memories are used commonly by all output lines.

Embodiment 3.

In the above embodiments 1 and 2, the number of input lines is one. The data queueing apparatus can be also configured for writing a plurality of data in parallel from a plurality of input lines 1a, 1b and 1c.

In Fig. 10, (a) shows an example of this embodiment. In the figure, the shift memories 4_1 - 4_9 are provided. The data is received at each of the input lines 1a, 1b, and 1c to each of the shift memories 4_7, 4_8, and 4_9. In the same way, each of the destination inputs is stored in each of the destination indicating bits In this example, the number of input lines is three and therefore three received data can be written in the shift memories in parallel simultaneously In this configuration, three shift memories store received data simultaneously as shown in Fig. 10 (b), therefore the apparatus shifts the data three times in one time slot. Namely, the apparatus is configured to receive data and to shift data three times in one time slot. In this embodiment, though the shift memory has to shift the data more quickly than Embodiments 1 and 2, the data can be received in the plurality of input lines to be processed simultaneously, so that a large amount of data can be processed at a high speed.

Embodiment 4.

In the above Embodiments 1 and 2, the search circuits are provided correspondingly to the output lines and the search circuits operate independently and simultaneously. But a data queueing apparatus can be also configured with fewer search circuits, each of which corresponds to a plurality of output lines. The search circuits operate at a high speed and each search circuit detects the data stored in the plurality of shift memories, so that the scale of the hardware can be reduced.

Fig. 11 shows an example of this embodiment. In the figure, the search circuit 6_1 is provided correspondingly to the output lines 2_1 and 2_2, and the search circuit 6_2 is provided correspondingly to the output lines 2_3 and 2_4. Namely, one search circuit is provided correspondingly to two output lines. The search circuits 6_1 and 6_2 operate at more than two times the speed of the search circuits in Embodiments 1 and 2, so that the same effect can be attained as in Embodiments 1 and 2.

Another case where only one search circuit is provided is possible (not shown in a figure) In this case, one search circuit is provided for the output lines 2_1 - 2_4, so that the search circuit needs to operate at more than four times the speed of the search circuits in Embodiments 1 and 2. The search circuit can be common in this case, and therefore the apparatus scale can be reduced.

Embodiment 5

In Embodiments 1 and 2 the shift memories shift the data one by one in one time slot, but it is desirable that the shifting speed is higher. The apparatus of this embodiment is configured as the same as in Embodiments 1 and 2, however, it shifts data asynchronous with the time slot. That is, the interval of shiftings is shorter than the period of the time slot. The speed of shifting data is increased, and data can be shifted forward more quickly. Thus, the received data is stored certainly in the shift memory.

Embodiment 6

In the above embodiments, the data is read once every four time slots, but in another case, for example, the data can be read once every time slot. This reading-out frequency can be vary.

Embodiment 7.

In the above embodiments, the speed of the input lines is the same as the speed of the output lines. However, a higher traffic concentration can be achieved by reading the data from the shift memory at a higher speed. It is also possible that the data can be written from the input line at a higher speed than it is read out at the output lines.

Embodiment 8

In the above embodiments, a set of the destination indicating bits and a search circuit are provided correspondingly to each of the output lines of the data queueing apparatus. A plurality of the above sets of the destination indicating bits and search circuits may be provided correspondingly to each of the delay priorities. The data having higher delay priority can be read first based on a code, which indicates the delay priority.

Fig. 12 shows an example of this embodiment. In the figure, a header converter 17, a first queueing apparatus 100, and a second queueing apparatus 200 are provided. In Fig. 12, two sets of the data queueing apparatuses are provided for one output line. The header converter 17 detects the delay priority included in the header of the received data. The data are distributed to the first queueing apparatus or the second queueing apparatus based on the delay priority. The first queueing apparatus is configured to have higher priority than the second queueing apparatus. For example, the first queueing apparatus may read the data once every four time slots and the second queueing apparatus may read the data once every eight time slots. By making such a difference between two queueing apparatuses, the priority of the first queueing apparatus becomes higher than the second queueing apparatus. Higher priority of the first queueing apparatus can also be got by reading the data in the second queueing apparatus after all data in the first queueing apparatus have been read.

Embodiment 9.

Fig. 13 shows another embodiment of this invention using priority. In this embodiment, priority can be controlled by indicating the destination with two bits instead of one bit. A priority table 30 and a count table 31 are included in the header converter 17 as shown in Fig. 13.

The priority table is a table showing a priority corresponding to two bits as shown in Fig. 14. The count table is a table for counting how many data items with different priorities are stored in the shift memories corresponding to each output lines, as shown in Fig. 15.

The header converter 17 receives the data having one of "high", "middle", "low" priorities and transmits it to the shift memory. The header converter 17 sets the value for each of destination indicating bits corresponding to the received priority. When the value of the bits is "00", no data exists. In other cases, the value of the bit is set correspondingly to each priority. The header converter 17 sets the priority with reference to the priority table 30.

The header converter 17 also increments the corresponding value of the count table 31 when the data and the destination indication is transmitted to the first shift memory. The count table shows how many data items, having each of priorities, exist for each of the output lines. The search circuit searches for and detects the data to be transferred to each of the output lines in an order of priority with reference to this count table. For example, the search circuit 6_1 is informed that one data having "high" priority and one data having "middle" priority exist by the count table 31 as shown in Fig. 15. The search circuit detects the data having "high" priority and requests to the selector to select and transfer it first. For example, in Fig. 13, the data having "high" priority is stored in the shift memory 4_2, so that the selector 7_1 selects and transfers the data stored in the shift memory 4_2 earlier than the data having "middle" priority stored in the shift memory 4_1 to the output line 2_1.

In this way, priority control can be processed by using plural bits in the destination indicating bits which show the priority of the data.

Embodiment 10

Furthermore, in the above embodiment, the data can be processed as a parallel signal by providing a serial-parallel converter and/or a parallel-serial converter in the pre-stage and/or the post-stage of the data queueing apparatus, if the operation speed is limited in the data queueing apparatus.

Fig. 16 shows this embodiment. In this case, the number of the input lines 1_y is eight. In this embodiment, the shift memory is configured to have eight bits and the input lines are provided correspondingly to each of bits, so that eight bits can be written in the shift memories in parallel. In this way, data transfer at a higher speed can be achieved by providing a plurality of the input lines compared with the case of one input line.

Embodiment 11.

In the above Embodiments 1 and 2, the limited-length data is received. The data can be address information in the packet switch, or the packet itself. Also other kinds of data can be processed with the apparatus described in the above Embodiments 1 and 2.

Embodiment 12.

In the above Embodiments 1 and 2, the shift memories and the destination indicating bits are configured as different memories. But the same memory unit can be used for the shift memories and the destination indicating bits as shown in Fig. 17. For example, when the shift memory has eight bits and the destination indicating bits have four bits, twelve bits are combined as one memory unit The received data and the destination inputs can be stored by the memory unit of twelve bits.

Embodiment 13.

The above Embodiments 1 and 2 show the apparatus in which the selectors are provided correspondingly to each output lines, but another apparatus can be configured with fewer selectors. For example, each selector may correspond to two output lines as shown in Fig. 18 In this embodiment, the selectors operate at twice the speed of the selectors in Embodiments 1 and 2, achieving the same effect as in the above embodiments can be attained

Embodiment 14.

The above Embodiment 4 and 13 show the apparatus in which the numbers of the search circuits and the selectors can be reduced by operating the search circuits and the selectors at a high speed. The numbers of the search circuits and the selectors can also be reduced even if they do not operate at a high speed as follows.

Fig. 19 shows the operation of the apparatus as configured as shown in Fig. 18. In the above Embodiments 1 and 2 (also in Embodiment 4 and 13), the data is transmitted every 4 time slots, but this embodiment shows the case in which the data is transmitted every time slot. The search circuit 6_1 searches and detects the existence of the data to be transferred to the output lines 2_1 and 2_2 alternately every time slot The selector 7_1 also transmits the data, if there is any, to the output lines 2_1 and 2_2 alternately every time slot. In this way, the search circuit and the selector do not need to operate at a high speed while detecting and transmitting the data every time slot. In Fig. 19, more effective operation can be achieved. The apparatus in this embodiment can transmit data every time slot, not every fourth slot, even if it operates at the same speed as in Embodiments 1 and 2.

Embodiment 15

Another apparatus can be provided using one search circuit and one selector as shown in Fig. 20. In this case, if the search circuit 6 and the selector 7 operate at four times the speed of the search circuits in Embodiments 1 and 2 the same effect can be attained as in the above embodiments

Embodiment 16.

Fig 21 shows the operation of the apparatus configured as shown in Fig 20, but in this case, the search circuit and the selector operate at the same speed as in Embodiments 1 and 2.

The search circuit detects and the selector transmits the data to be transferred to each output line once every time slot in the case shown in Fig. 21. Therefore, the same effect can be attained by detecting and selecting the data to be transferred to each output line once every time slot. In this case, the apparatus needs only one search circuit 6 and one selector 7, in addition, they do not need to operate at a high speed, so that the circuits are simply configured

Embodiment 17.

Fig 22 shows an example of a shared multibuffer ATM switch. This shared multibuffer ATM switch is characterized by adopting in the controller 16 the data queueing apparatus 180 described above

Fig. 23 shows a configuration of the data queueing apparatus 180. The data queueing apparatus 180 has the same function as the data queueing apparatus described in Embodiment 1. Flags correspond to the above-described destination indicating bits which store the destination information. Address corresponds to the data stored in the above-described shift memories. And a searching element corresponds to the above-described search circuit.

Fig. 24 shows a shift operation The value of an occupation indicator becomes "0" when all of the flags are "0". When the occupation indicator indicates "0", the data stored in the previous stage is shifted to the next

Fig 25 explains the forward shifting shown in Fig. 24. In the first stage, the flags have been set 1, 0, 0, so that the occupation indicator indicates "1" as shown in Fig. 25. The value of the occupation indicator is transmitted to a shift trigger of the second stage The shift trigger shifts the data which are stored in the stage where the shift trigger exists, to the next stage when the value of the occupation indicator becomes "0". In the second stage, the flags are all "0", so that the occupation indicator indicates "0". The shift trigger of the third stage shifts the address and the destination indication, stored in the third stage, to the second stage.

Fig. 26 is a graph comparing the memory size needed by the conventional data queueing apparatus and by the data queueing apparatus according to this embodiment. In Fig. 26, the horizontal axis shows the number of the output lines and the vertical axis shows the ratio of the memory size needed by the apparatus according to this embodiment to the memory size needed by the conventional apparatus The graph shown in Fig. 26 is made using the formula as shown in Fig 27. As shown in Fig. 26, the more output ports the apparatus has, the smaller memory size is needed by the data queueing apparatus of this embodiment compared with the conventional apparatus.

Embodiment 18.

Fig. 28 shows another embodiment for shifting the data stored in the shift memories. In the above embodiments, the data stored in the shift memories are shifted forward once every time slot. It is desirable to shift the stored data a plurality of times every time slot as shown in Fig 28, not to shift data once every time slot. In this case, data are received every time slot, and the received data can be certainly stored in the shift memories by previously shifting forward the stored data.

Embodiment 19.

Fig. 29 shows another example of the searching element. In the above embodiment, the searching element is provided correspondingly to each output line. In this embodiment, the searching element is provided in common for a plurality of the output lines. A switch 60 is provided for the searching element, and the switch changes the connections with the flags for searching each flag sequentially. In this case, the searching element may operate at a high speed, or the searching element may not operate at a high speed for searching the data to be transmitted to each output line once every time slot. In this embodiment, the same effect can be attained as in the above embodiments.

Embodiment 20

Each of the flags is not needed to be 1 bit, but it can consist of 2 or more bits, not shown in the figure. When a flag has 2 or more bits, the priorities of the data can be processed.

The invention has been described in connection with a number of specific embodiments. These embodiments are given by way of example only, and the scope of the invention is limited only by the scope of the appended claims.

Claims

1. A data queueing apparatus comprising:
   (a) a plurality of output lines for transmitting data;
   (b) input means for receiving data with destination information which indicates at least one of the output lines;
   (c) memory means, which is provided for the plurality of output lines, for storing data received at the input means; and
   (d) search and select means which searches outgoing data for each output lines among the data stored in the memory means with the destination information, and selects the searched data in the memory means for the corresponding output line for transmitting.

2. The data queueing apparatus of claim 1, wherein the memory means comprises:
   (a) a plurality of shift memories for shifting data and storing data therein;
   (b) a plurality of destination memories each of which corresponds to both each shift memory and each output line and stores the destination information of data therein.

3. The data queueing apparatus of claim 2, wherein the search and select means comprises a plurality of search means each of which corresponds to each of the plurality of the output lines and searches destination information stored in the corresponding destination memories for finding data to transmit to the corresponding output line.

4. The data queueing apparatus of claim 2, wherein the search and select means comprises a plurality of select means each of which corresponds to each of the plurality of output lines and selects data found by the search means and transmits data to the corresponding output lines.

5. The data queueing apparatus of claim 1, further comprising delete means for deleting data stored in the memory means after being selected and transmitted by the search and select means

6. The data queueing apparatus of claim 5, wherein input means receives data with destination information, indicative of destination output lines, and the delete means deletes data after they are transmitted to the plurality of output lines indicated by the destination information.

7. The data queueing apparatus of claim 2, further comprising a plurality of shift detectors, each of which corresponds to each of the shift memories, for requesting a previous shift memory to shift data when it detects selecting and transmitting of the data stored in the corresponding shift memory

8. The data queueing apparatus of claim 2, wherein the input means expects to receive data every time slot, which has predefined time length, the plurality of shift memories shift data more than once every time slot, and the search and select means searches data once every predefined X (X> =2) time slot.

9. The data queueing apparatus of claim 2, wherein the input means expects to receive data each time slot which has a predefined time length, includes Y (Y> =2) number of input lines which are coupled to the Y number of shift memories respectively for receiving data once to the coupled shift memory for every time slot, and the plurality of shift memories shift data Y number of times for every time slot.

10. The data queueing apparatus of claim 3, wherein the search and select means comprises search means whose number is less than the number of output lines.

11. The data queueing apparatus of claim 4, wherein the search and select means comprises select means whose number is less than the number of output lines

12. The data queueing apparatus of claim 1, wherein the input means receives a serial signal and includes serial-parallel converter for transforming the serial signal to the parallel signal, and the memory means receives the parallel signal from the serial-parallel converter

13. The data queueing apparatus of claim 2, wherein the shift memory and the destination memory are coupled together as a memory unit.

14. A data queueing system comprising;

(a) first data queueing apparatus of claim 1 for processing data having priority;

(b) second data queueing apparatus of claim 1 for processing data having different priority;

(c) data distributing means for inputting data with a priority and transferring data to one of the first and second data queueing apparatuses based on the priority of received data.

15. The data queueing apparatus of claim 2, wherein the input means receives the destination information, along with priority for processing data, the memory means stores the destination information and the priority, and the search and select means searches the destination information for selecting data based on the stored priority.

16. A data queueing apparatus comprising:

(a) an input line for receiving a limited length of data defined by a predefined fixed time;

(b) a plurality of output lines for transmitting a limited length of data;

(c) destination input means, which corresponds to the input line, for receiving destination information which indicates one of the plurality of output lines corresponding to the data;

(d) a plurality of shift memories, which are connected serially, for storing received data and shifting stored data forward serially;

(e) a plurality of destination indicating bits, each of which corresponds to both of each of the shift memories and each of the output lines, and is asserted when the bit corresponds to a destination of data stored in one of the shift memories, for storing the destination received at the destination input means, and shifting the stored destination synchronous with the shifting of the data stored in the corresponding shift memories;

(f) search circuits, each of which corresponds to each of the output lines, for searching backward for an asserted bit in the destination indicating bits; and

(g) selectors for selecting data from the plurality of shift memories to the output lines based on search results of the search circuits.

17. The data queueing apparatus of claim 16, wherein one of a limited length of data is transmitted to the plurality of output lines, further comprising shift detectors, each of which corresponds to each of the shift memories, for detecting that all of the destination indicating bits which correspond to one of the shift memories are not asserted and directing next shift memory and next destination indicating bits to shift data and the destination forward.

18. A data queueing method comprising the steps of:

    (a) receiving data with destination information from an input line every time slot;
    (b) storing data with the destination information in a common memory in an order;
    (c) searching the destination information in the order for destinations corresponding to each of a plurality of output lines;
    (d) selecting data for each of the plurality of output lines based on a search result;
    (e) transmitting selected data to each of output lines.

19. The data queueing method of claim 18, wherein the receiving step comprises the steps of:

    (a) checking if the common memory is full of data; and
    (b) discarding data received if the common memory is full of data.

20. The data queueing method of claim 18, further comprising the step of marking as deleted that data transmitted from the common memory after the transmitting step

21. The data queueing method of claim 20, wherein the storing step comprises the steps of

    (a) checking if the common memory holds the data marked as deleted;
    (b) shifting other data in the common memory onto the data marked as deleted if the common memory holds the data marked as deleted;

22. The data queueing method of claim 18, wherein the searching step comprises the step of repeating the receiving step and the storing step a plurality of times before searching the destination information.

23. An Asynchronous Transfer Mode switch for switching cells, having destination information therewith, between input ports and output ports, comprising:

    (a) a shared multibuffer having addresses for storing cells;
    (b) a controller having a data queueing apparatus as recited in claim 1, providing the addresses, whereat the cells are stored in the shared multibuffer with the destinations of the stored cells to the data queueing apparatus, whereby the data queueing apparatus transmits the addresses of cells to be transmitted to the output ports

24. An Asynchronous Transfer Mode switch for switching cells, having destination information therewith between input ports and output ports, comprising:

    (a) a shared multibuffer having addresses for storing cells;
    (b) a controller having a common address queue for queueing the addresses of cells stored in the shared multibuffer and the destination information.

25. The Asynchronous Transfer Mode switch of claim 24, wherein the address queue comprises memory means for storing the addresses and the destination information of cells, and select means for selecting addresses for each of output ports based on the destination information.

26. The Asynchronous Transfer Mode switch of claim 25, wherein the memory means comprises shift memories for storing addresses and destination information, and searching elements for searching destination information stored in the shift memories.

27. The Asynchronous Transfer Mode switch of claim 26, wherein each of the shift memories comprises occupation indicator for indicating an occupation status of the shift memory, and shift trigger for shifting addresses and destination information to another shift memory based on the occupation status indicated by the occupation indicator in the other shift memory.

FIG. 1

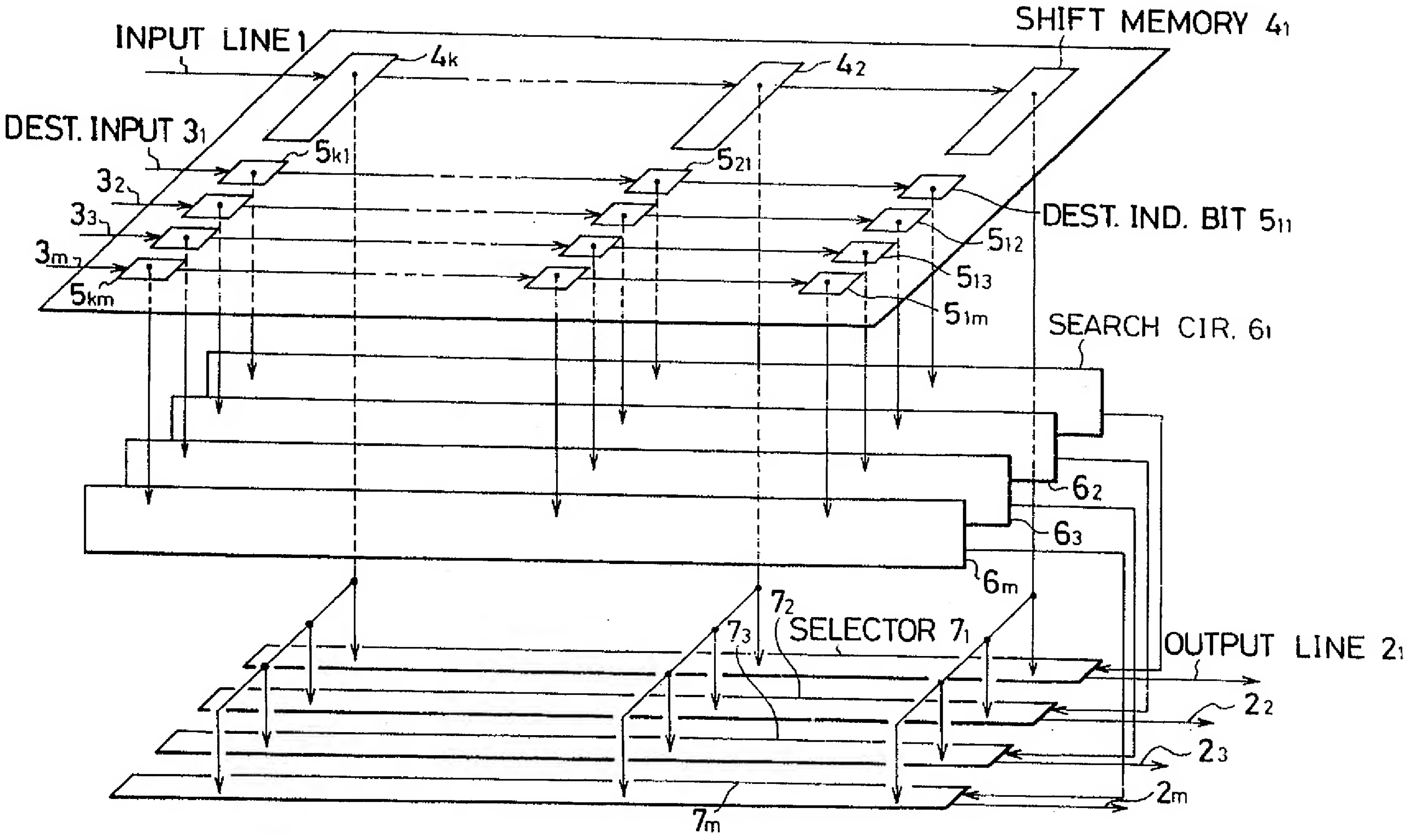
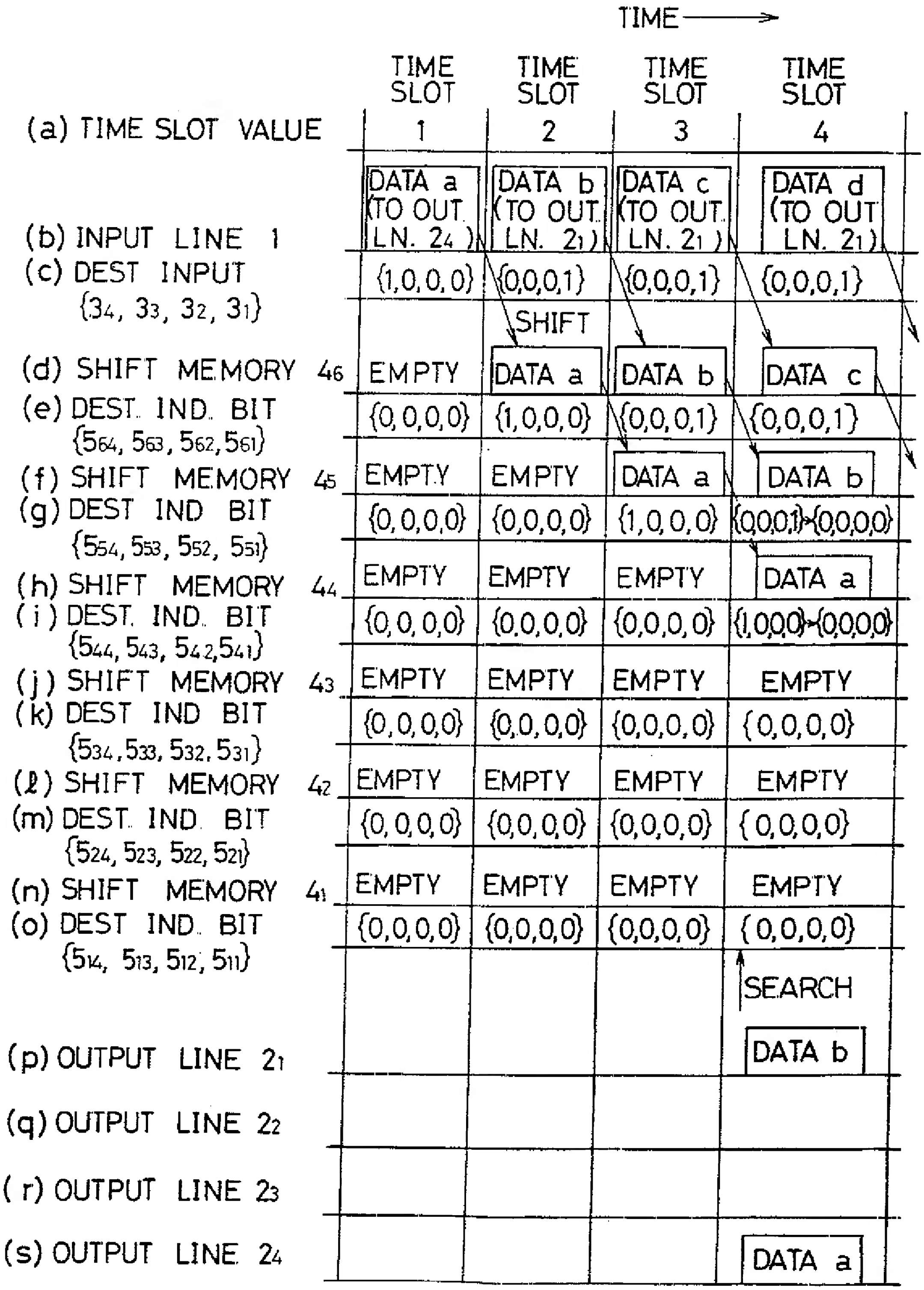

INPUT LINE 1
SHIFT MEMORY 4_1
4_k
4_2
DEST. INPUT 3_1
3_2
3_3
3_m
5_{k1}
5_{km}
5_{21}
DEST. IND. BIT 5_{11}
5_{12}
5_{13}
5_{1m}
SEARCH CIR. 6_1
6_2
6_3
6_m
7_2
7_3
SELECTOR 7_1
7_m
OUTPUT LINE 2_1
2_2
2_3
2_m

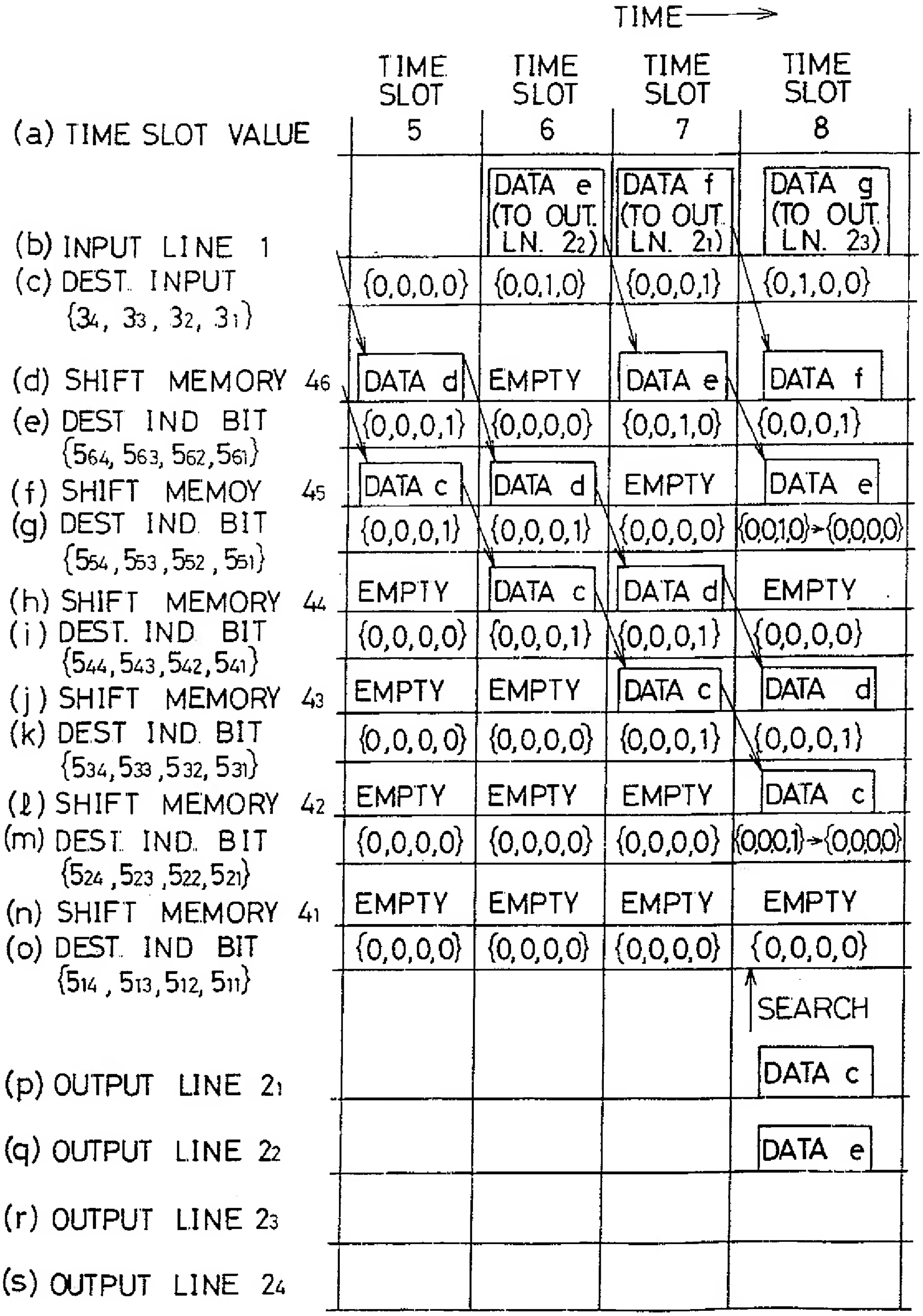
FIG. 2



TIME →

		TIME SLOT	TIME SLOT	TIME SLOT	TIME SLOT
(a) TIME SLOT VALUE		1	2	3	4
(b) INPUT LINE 1		DATA a (TO OUT. LN. 2_4)	DATA b (TO OUT. LN. 2_1)	DATA c (TO OUT. LN. 2_1)	DATA d (TO OUT LN. 2_1)
(c) DEST INPUT {3_4, 3_3, 3_2, 3_1}		{1,0,0,0}	{0,0,0,1}	{0,0,0,1}	{0,0,0,1}
			SHIFT		
(d) SHIFT MEMORY	4_6	EMPTY	DATA a	DATA b	DATA c
(e) DEST. IND. BIT {5_{64}, 5_{63}, 5_{62}, 5_{61}}		{0,0,0,0}	{1,0,0,0}	{0,0,0,1}	{0,0,0,1}
(f) SHIFT MEMORY	4_5	EMPTY	EMPTY	DATA a	DATA b
(g) DEST IND BIT {5_{54}, 5_{53}, 5_{52}, 5_{51}}		{0,0,0,0}	{0,0,0,0}	{1,0,0,0}	{0,0,0,1}→{0,0,0,0}
(h) SHIFT MEMORY	4_4	EMPTY	EMPTY	EMPTY	DATA a
(i) DEST. IND. BIT {5_{44}, 5_{43}, 5_{42}, 5_{41}}		{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{1,0,0,0}→{0,0,0,0}
(j) SHIFT MEMORY	4_3	EMPTY	EMPTY	EMPTY	EMPTY
(k) DEST IND BIT {5_{34}, 5_{33}, 5_{32}, 5_{31}}		{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}
(ℓ) SHIFT MEMORY	4_2	EMPTY	EMPTY	EMPTY	EMPTY
(m) DEST. IND. BIT {5_{24}, 5_{23}, 5_{22}, 5_{21}}		{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}
(n) SHIFT MEMORY	4_1	EMPTY	EMPTY	EMPTY	EMPTY
(o) DEST. IND. BIT {5_{14}, 5_{13}, 5_{12}, 5_{11}}		{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}
					SEARCH
(p) OUTPUT LINE 2_1					DATA b
(q) OUTPUT LINE 2_2					
(r) OUTPUT LINE 2_3					
(s) OUTPUT LINE 2_4					DATA a

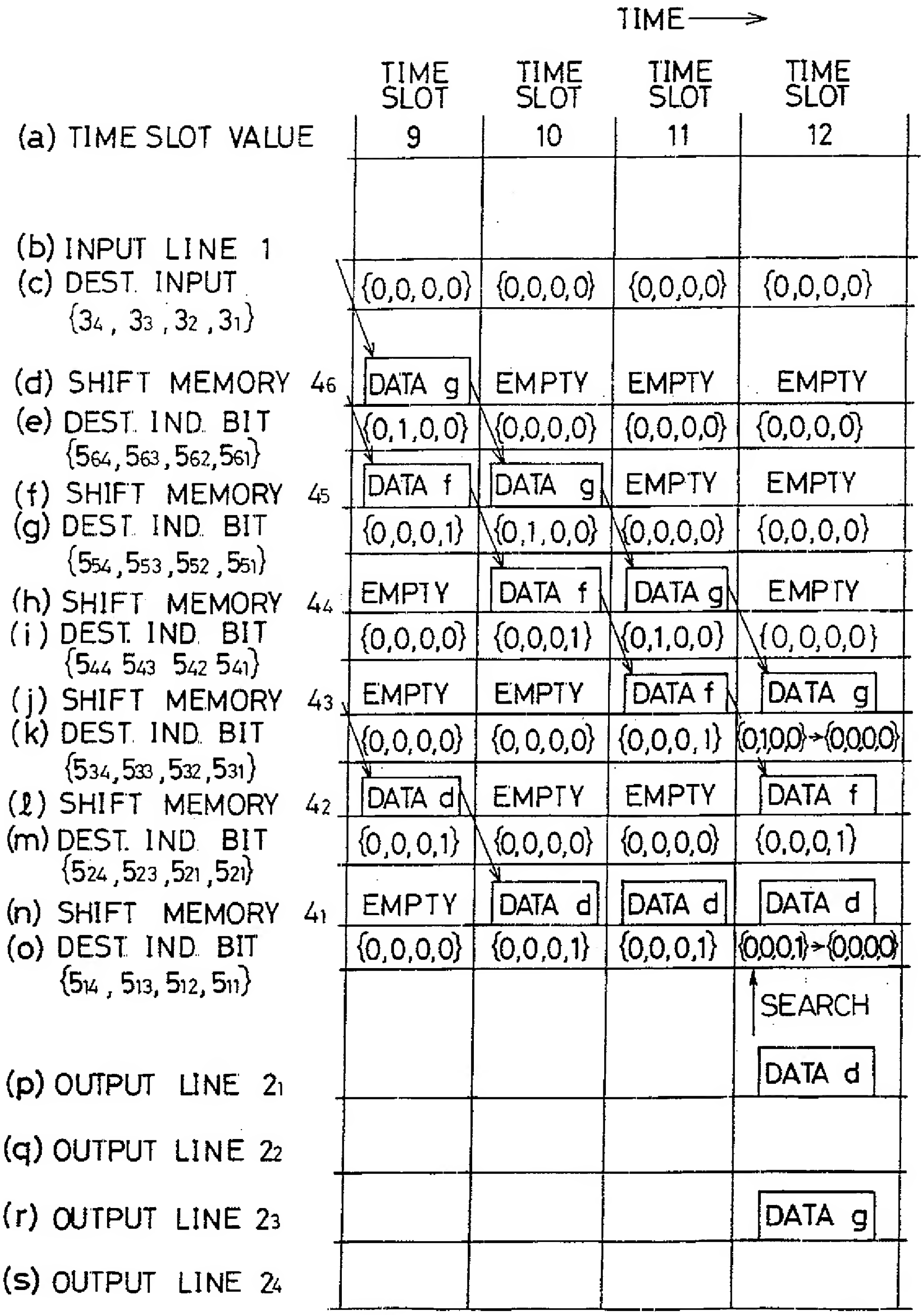
FIG. 3



TIME →

	TIME SLOT	TIME SLOT	TIME SLOT	TIME SLOT
(a) TIME SLOT VALUE	5	6	7	8
(b) INPUT LINE 1		DATA e (TO OUT. LN. 2_2)	DATA f (TO OUT. LN. 2_1)	DATA g (TO OUT. LN. 2_3)
(c) DEST. INPUT $\{3_4, 3_3, 3_2, 3_1\}$	{0,0,0,0}	{0,0,1,0}	{0,0,0,1}	{0,1,0,0}
(d) SHIFT MEMORY 46	DATA d	EMPTY	DATA e	DATA f
(e) DEST IND BIT $\{5_{64}, 5_{63}, 5_{62}, 5_{61}\}$	{0,0,0,1}	{0,0,0,0}	{0,0,1,0}	{0,0,0,1}
(f) SHIFT MEMOY 45	DATA c	DATA d	EMPTY	DATA e
(g) DEST IND. BIT $\{5_{54}, 5_{53}, 5_{52}, 5_{51}\}$	{0,0,0,1}	{0,0,0,1}	{0,0,0,0}	{0,0,1,0}→{0,0,0,0}
(h) SHIFT MEMORY 44	EMPTY	DATA c	DATA d	EMPTY
(i) DEST. IND. BIT $\{5_{44}, 5_{43}, 5_{42}, 5_{41}\}$	{0,0,0,0}	{0,0,0,1}	{0,0,0,1}	{0,0,0,0}
(j) SHIFT MEMORY 43	EMPTY	EMPTY	DATA c	DATA d
(k) DEST IND. BIT $\{5_{34}, 5_{33}, 5_{32}, 5_{31}\}$	{0,0,0,0}	{0,0,0,0}	{0,0,0,1}	{0,0,0,1}
(ℓ) SHIFT MEMORY 42	EMPTY	EMPTY	EMPTY	DATA c
(m) DEST. IND. BIT $\{5_{24}, 5_{23}, 5_{22}, 5_{21}\}$	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,1}→{0,0,0,0}
(n) SHIFT MEMORY 41	EMPTY	EMPTY	EMPTY	EMPTY
(o) DEST. IND BIT $\{5_{14}, 5_{13}, 5_{12}, 5_{11}\}$	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}
(p) OUTPUT LINE 2_1				↑SEARCH DATA c
(q) OUTPUT LINE 2_2				DATA e
(r) OUTPUT LINE 2_3				
(s) OUTPUT LINE 2_4				

F I G. 4



TIME ⟶

	TIME SLOT	TIME SLOT	TIME SLOT	TIME SLOT
(a) TIME SLOT VALUE	9	10	11	12
(b) INPUT LINE 1				
(c) DEST. INPUT {3_4, 3_3, 3_2, 3_1}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}
(d) SHIFT MEMORY 4_6	DATA g	EMPTY	EMPTY	EMPTY
(e) DEST. IND. BIT {5_{64},5_{63},5_{62},5_{61}}	{0,1,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}
(f) SHIFT MEMORY 4_5	DATA f	DATA g	EMPTY	EMPTY
(g) DEST. IND. BIT {5_{54},5_{53},5_{52},5_{51}}	{0,0,0,1}	{0,1,0,0}	{0,0,0,0}	{0,0,0,0}
(h) SHIFT MEMORY 4_4	EMPTY	DATA f	DATA g	EMPTY
(i) DEST. IND. BIT {5_{44} 5_{43} 5_{42} 5_{41}}	{0,0,0,0}	{0,0,0,1}	{0,1,0,0}	{0,0,0,0}
(j) SHIFT MEMORY 4_3	EMPTY	EMPTY	DATA f	DATA g
(k) DEST. IND. BIT {5_{34},5_{33},5_{32},5_{31}}	{0,0,0,0}	{0,0,0,0}	{0,0,0,1}	{0,1,0,0}→{0,0,0,0}
(ℓ) SHIFT MEMORY 4_2	DATA d	EMPTY	EMPTY	DATA f
(m) DEST. IND. BIT {5_{24},5_{23},5_{21},5_{21}}	{0,0,0,1}	{0,0,0,0}	{0,0,0,0}	{0,0,0,1}
(n) SHIFT MEMORY 4_1	EMPTY	DATA d	DATA d	DATA d
(o) DEST. IND. BIT {5_{14}, 5_{13}, 5_{12}, 5_{11}}	{0,0,0,0}	{0,0,0,1}	{0,0,0,1}	{0,0,0,1}→{0,0,0,0}
				SEARCH
(p) OUTPUT LINE 2_1				DATA d
(q) OUTPUT LINE 2_2				
(r) OUTPUT LINE 2_3				DATA g
(s) OUTPUT LINE 2_4				

FIG. 5

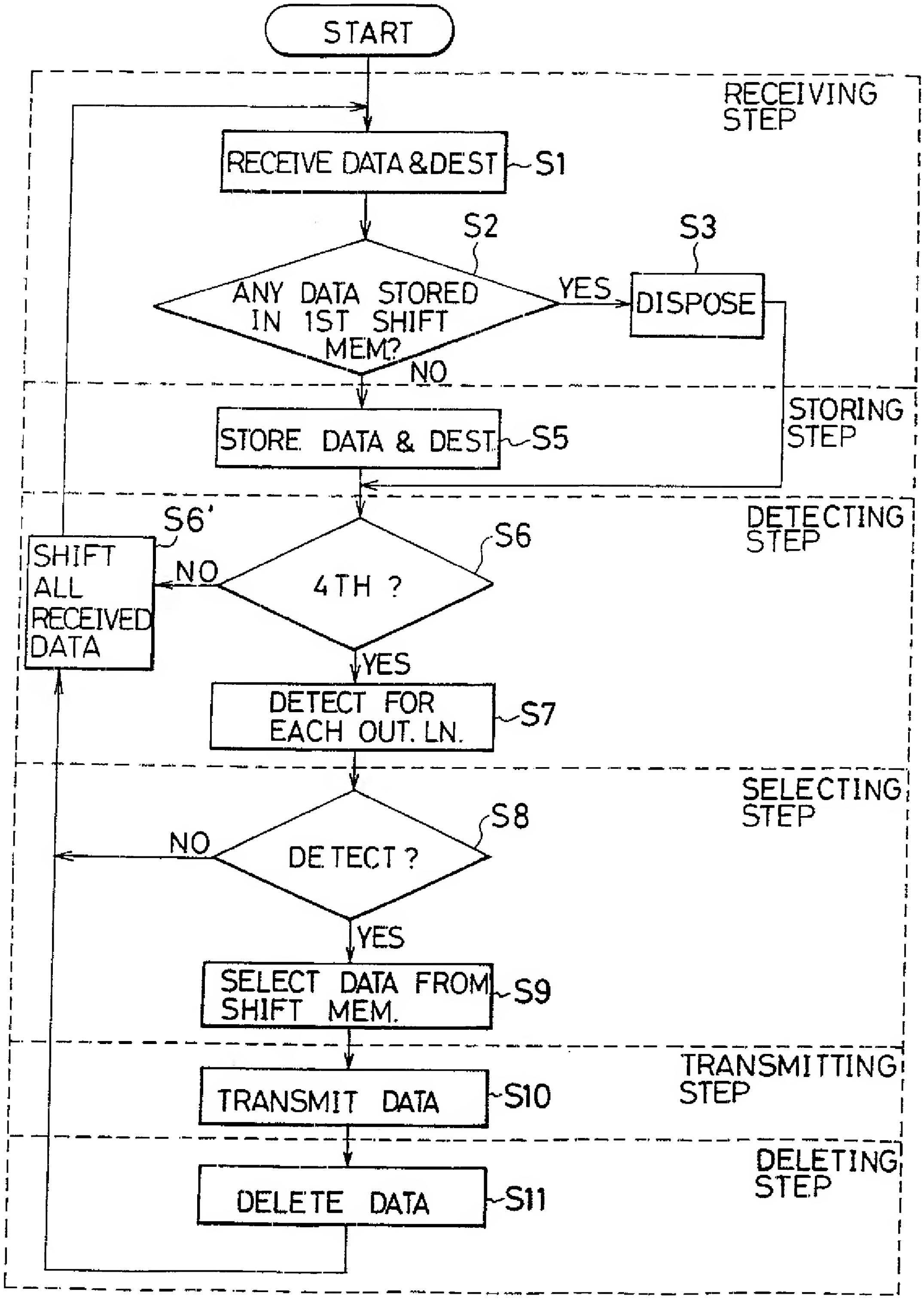

START
RECEIVING STEP
RECEIVE DATA & DEST
S1
S2
ANY DATA STORED IN 1ST SHIFT MEM?
YES
S3
DISPOSE
NO
STORING STEP
STORE DATA & DEST
S5
DETECTING STEP
S6'
SHIFT ALL RECEIVED DATA
NO
4TH ?
S6
YES
DETECT FOR EACH OUT. LN.
S7
SELECTING STEP
S8
NO
DETECT ?
YES
SELECT DATA FROM SHIFT MEM.
S9
TRANSMITTING STEP
TRANSMIT DATA
S10
DELETING STEP
DELETE DATA
S11

FIG. 6

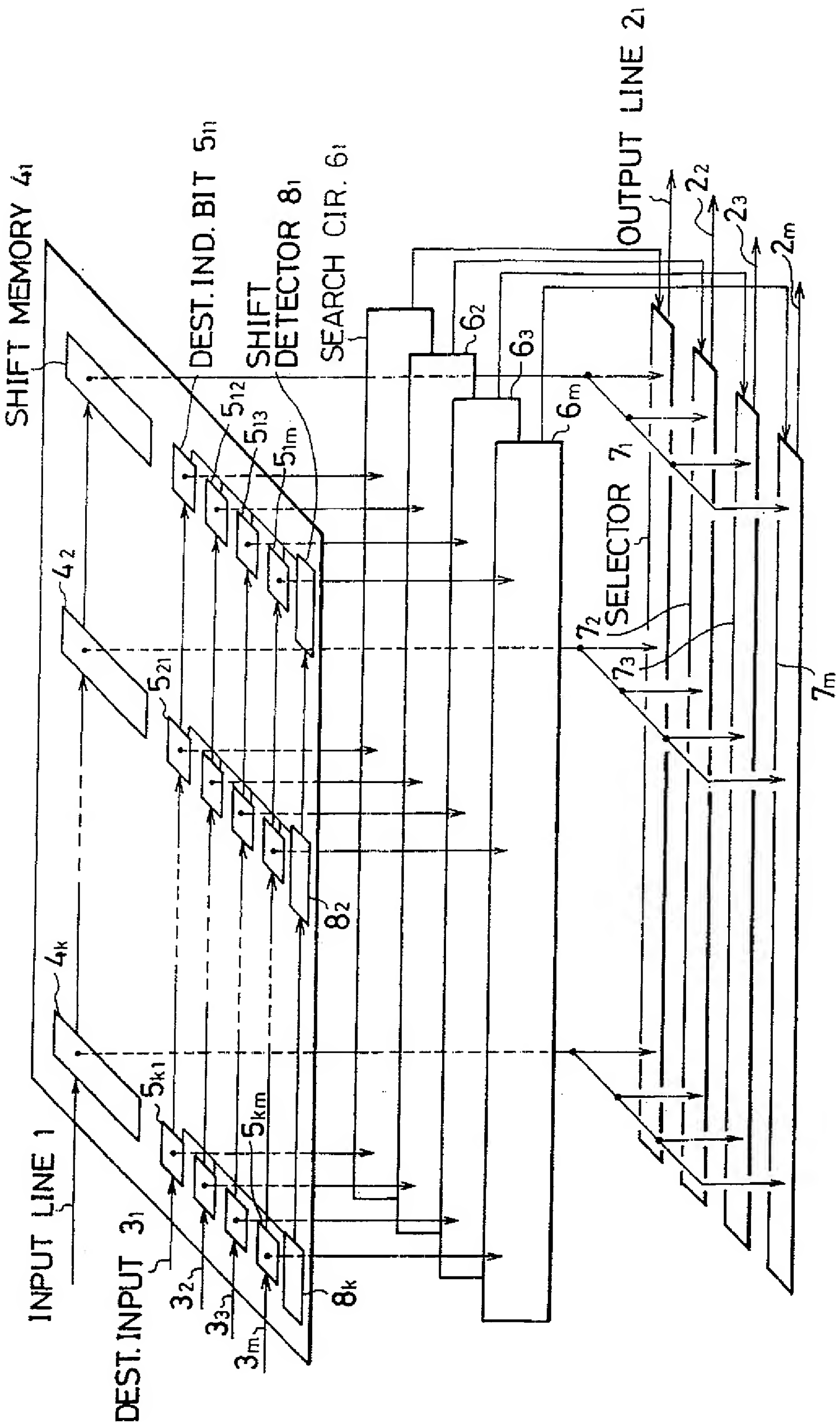
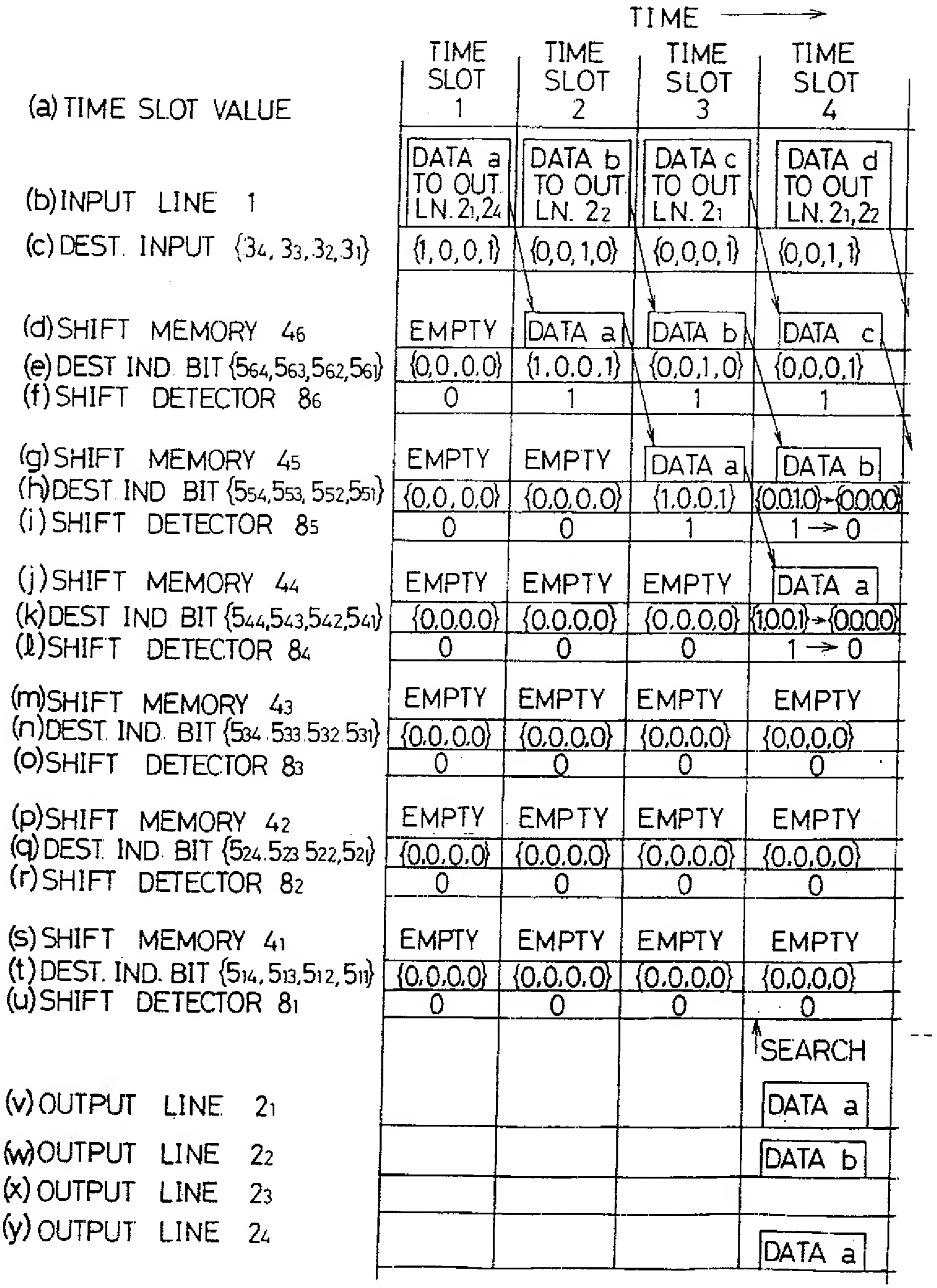

SHIFT MEMORY 4_1
DEST. IND. BIT 5_{11}
SHIFT DETECTOR 8_1
SEARCH CIR. 6_1
OUTPUT LINE 2_1
2_2
2_3
2_m
6_2
6_3
6_m
5_{12}
5_{13}
5_{1m}
SELECTOR 7_1
4_2
5_{21}
7_2
7_3
7_m
8_2
4_k
5_{k1}
5_{km}
INPUT LINE 1
DEST. INPUT 3_1
3_2
3_3
3_m
8_k

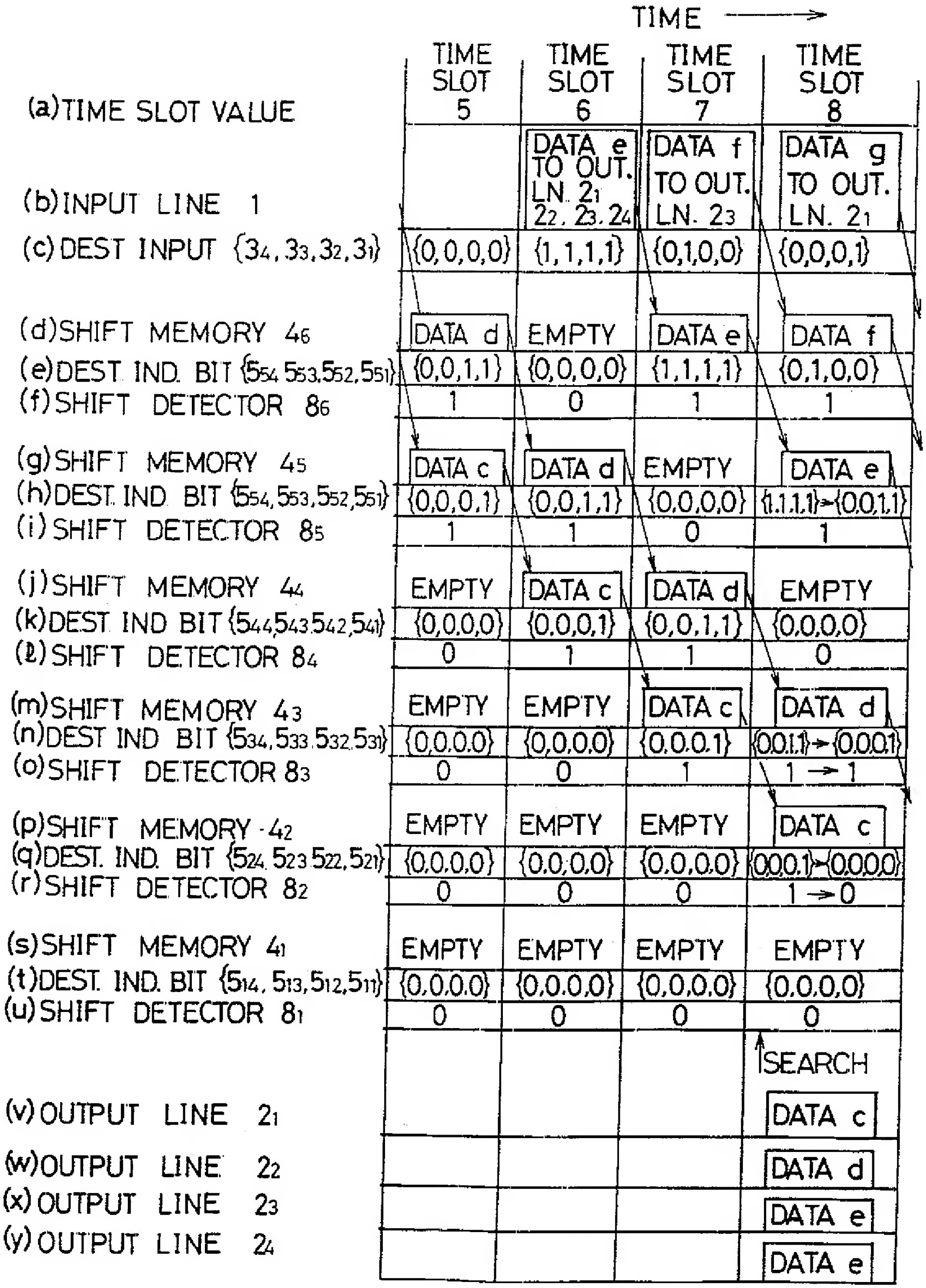
FIG 7



TIME →

(a) TIME SLOT VALUE	TIME SLOT 1	TIME SLOT 2	TIME SLOT 3	TIME SLOT 4
(b) INPUT LINE 1	DATA a TO OUT LN. 2_1,2_4	DATA b TO OUT LN. 2_2	DATA c TO OUT LN. 2_1	DATA d TO OUT LN. 2_1,2_2
(c) DEST. INPUT {3_4, 3_3, 3_2, 3_1}	{1, 0, 0, 1}	{0, 0, 1, 0}	{0, 0, 0, 1}	{0, 0, 1, 1}
(d) SHIFT MEMORY 4_6	EMPTY	DATA a	DATA b	DATA c
(e) DEST IND. BIT {5_{64},5_{63},5_{62},5_{61}}	{0,0,0,0}	{1,0,0,1}	{0,0,1,0}	{0,0,0,1}
(f) SHIFT DETECTOR 8_6	0	1	1	1
(g) SHIFT MEMORY 4_5	EMPTY	EMPTY	DATA a	DATA b
(h) DEST. IND BIT {5_{54},5_{53}, 5_{52},5_{51}}	{0,0, 0,0}	{0,0,0,0}	{1,0,0,1}	{0,0,1,0}→{0,0,0,0}
(i) SHIFT DETECTOR 8_5	0	0	1	1 → 0
(j) SHIFT MEMORY 4_4	EMPTY	EMPTY	EMPTY	DATA a
(k) DEST IND BIT {5_{44},5_{43},5_{42},5_{41}}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{1,0,0,1}→{0,0,0,0}
(ℓ) SHIFT DETECTOR 8_4	0	0	0	1 → 0
(m) SHIFT MEMORY 4_3	EMPTY	EMPTY	EMPTY	EMPTY
(n) DEST. IND. BIT {5_{34}.5_{33}.5_{32}.5_{31}}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}
(o) SHIFT DETECTOR 8_3	0	0	0	0
(p) SHIFT MEMORY 4_2	EMPTY	EMPTY	EMPTY	EMPTY
(q) DEST. IND. BIT {5_{24}.5_{23} 5_{22},5_{21}}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}
(r) SHIFT DETECTOR 8_2	0	0	0	0
(s) SHIFT MEMORY 4_1	EMPTY	EMPTY	EMPTY	EMPTY
(t) DEST. IND. BIT {5_{14}, 5_{13},5_{12}, 5_{11}}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}
(u) SHIFT DETECTOR 8_1	0	0	0	0
				↑SEARCH
(v) OUTPUT LINE 2_1				DATA a
(w) OUTPUT LINE 2_2				DATA b
(x) OUTPUT LINE 2_3				
(y) OUTPUT LINE 2_4				DATA a

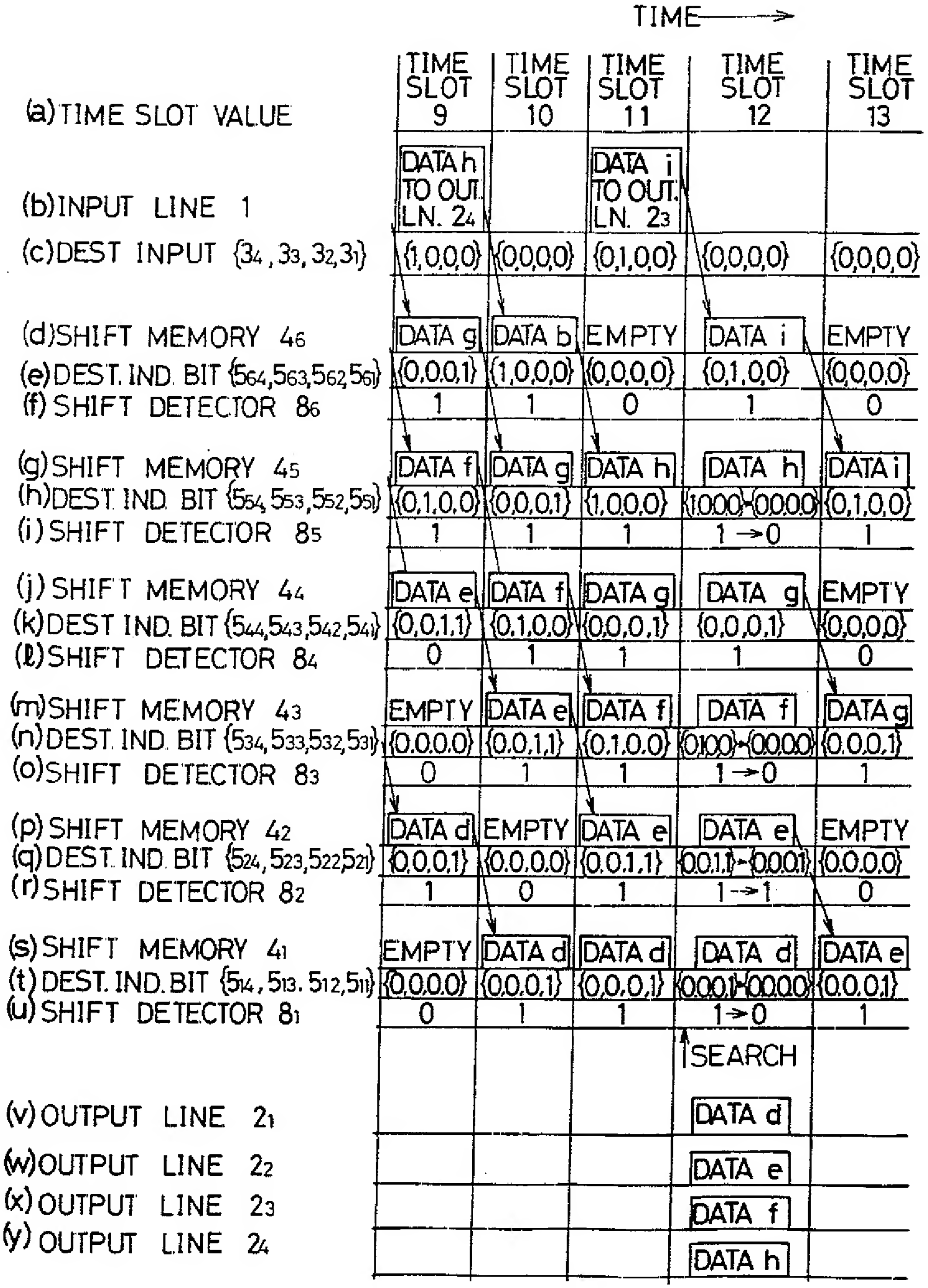
FIG. 8



TIME ⟶

(a) TIME SLOT VALUE	TIME SLOT 5	TIME SLOT 6	TIME SLOT 7	TIME SLOT 8
(b) INPUT LINE 1		DATA e TO OUT. LN. 2_1 2_2, 2_3, 2_4	DATA f TO OUT. LN. 2_3	DATA g TO OUT. LN. 2_1
(c) DEST INPUT $\{3_4, 3_3, 3_2, 3_1\}$	{0, 0, 0, 0}	{1, 1, 1, 1}	{0,1,0,0}	{0,0,0,1}
(d) SHIFT MEMORY 4_6	DATA d	EMPTY	DATA e	DATA f
(e) DEST. IND. BIT $\{5_{54}, 5_{53}, 5_{52}, 5_{51}\}$	{0,0,1,1}	{0,0,0,0}	{1,1,1,1}	{0,1,0,0}
(f) SHIFT DETECTOR 8_6	1	0	1	1
(g) SHIFT MEMORY 4_5	DATA c	DATA d	EMPTY	DATA e
(h) DEST. IND. BIT $\{5_{54}, 5_{53}, 5_{52}, 5_{51}\}$	{0,0,0,1}	{0,0,1,1}	{0,0,0,0}	{1,1,1,1}→{0,0,1,1}
(i) SHIFT DETECTOR 8_5	1	1	0	1
(j) SHIFT MEMORY 4_4	EMPTY	DATA c	DATA d	EMPTY
(k) DEST IND BIT $\{5_{44}, 5_{43}, 5_{42}, 5_{41}\}$	{0,0,0,0}	{0,0,0,1}	{0,0,1,1}	{0,0,0,0}
(ℓ) SHIFT DETECTOR 8_4	0	1	1	0
(m) SHIFT MEMORY 4_3	EMPTY	EMPTY	DATA c	DATA d
(n) DEST IND BIT $\{5_{34}, 5_{33}, 5_{32}, 5_{31}\}$	{0,0,0,0}	{0,0,0,0}	{0,0,0,1}	{0,0,1,1}→{0,0,0,1}
(o) SHIFT DETECTOR 8_3	0	0	1	1 → 1
(p) SHIFT MEMORY - 4_2	EMPTY	EMPTY	EMPTY	DATA c
(q) DEST. IND. BIT $\{5_{24}, 5_{23}, 5_{22}, 5_{21}\}$	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,1}→{0,0,0,0}
(r) SHIFT DETECTOR 8_2	0	0	0	1 → 0
(s) SHIFT MEMORY 4_1	EMPTY	EMPTY	EMPTY	EMPTY
(t) DEST. IND. BIT $\{5_{14}, 5_{13}, 5_{12}, 5_{11}\}$	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}
(u) SHIFT DETECTOR 8_1	0	0	0	0
				SEARCH
(v) OUTPUT LINE 2_1				DATA c
(w) OUTPUT LINE 2_2				DATA d
(x) OUTPUT LINE 2_3				DATA e
(y) OUTPUT LINE 2_4				DATA e

FIG. 9



TIME →

(a) TIME SLOT VALUE	TIME SLOT 9	TIME SLOT 10	TIME SLOT 11	TIME SLOT 12	TIME SLOT 13
(b) INPUT LINE 1	DATA h TO OUT. LN. 2_4		DATA i TO OUT. LN. 2_3		
(c) DEST INPUT $\{3_4, 3_3, 3_2, 3_1\}$	{1,0,0,0}	{0,0,0,0}	{0,1,0,0}	{0,0,0,0}	{0,0,0,0}
(d) SHIFT MEMORY 4_6	DATA g	DATA b	EMPTY	DATA i	EMPTY
(e) DEST. IND. BIT $\{5_{64}, 5_{63}, 5_{62}, 5_{61}\}$	{0,0,0,1}	{1,0,0,0}	{0,0,0,0}	{0,1,0,0}	{0,0,0,0}
(f) SHIFT DETECTOR 8_6	1	1	0	1	0
(g) SHIFT MEMORY 4_5	DATA f	DATA g	DATA h	DATA h	DATA i
(h) DEST. IND. BIT $\{5_{54}, 5_{53}, 5_{52}, 5_{51}\}$	{0,1,0,0}	{0,0,0,1}	{1,0,0,0}	{1000}→{0000}	{0,1,0,0}
(i) SHIFT DETECTOR 8_5	1	1	1	1 → 0	1
(j) SHIFT MEMORY 4_4	DATA e	DATA f	DATA g	DATA g	EMPTY
(k) DEST IND. BIT $\{5_{44}, 5_{43}, 5_{42}, 5_{41}\}$	{0,0,1,1}	{0,1,0,0}	{0,0,0,1}	{0,0,0,1}	{0,0,0,0}
(ℓ) SHIFT DETECTOR 8_4	0	1	1	1	0
(m) SHIFT MEMORY 4_3	EMPTY	DATA e	DATA f	DATA f	DATA g
(n) DEST. IND. BIT $\{5_{34}, 5_{33}, 5_{32}, 5_{31}\}$	{0,0,0,0}	{0,0,1,1}	{0,1,0,0}	{0100}→{0000}	{0,0,0,1}
(o) SHIFT DETECTOR 8_3	0	1	1	1 → 0	1
(p) SHIFT MEMORY 4_2	DATA d	EMPTY	DATA e	DATA e	EMPTY
(q) DEST. IND. BIT $\{5_{24}, 5_{23}, 5_{22}, 5_{21}\}$	{0,0,0,1}	{0,0,0,0}	{0,0,1,1}	{0011}→{0001}	{0,0,0,0}
(r) SHIFT DETECTOR 8_2	1	0	1	1 → 1	0
(s) SHIFT MEMORY 4_1	EMPTY	DATA d	DATA d	DATA d	DATA e
(t) DEST. IND. BIT $\{5_{14}, 5_{13}, 5_{12}, 5_{11}\}$	{0,0,0,0}	{0,0,0,1}	{0,0,0,1}	{0001}→{0000}	{0,0,0,1}
(u) SHIFT DETECTOR 8_1	0	1	1	1 → 0	1
				↑SEARCH	
(v) OUTPUT LINE 2_1				DATA d	
(w) OUTPUT LINE 2_2				DATA e	
(x) OUTPUT LINE 2_3				DATA f	
(y) OUTPUT LINE 2_4				DATA h	

FIG. 10(a)

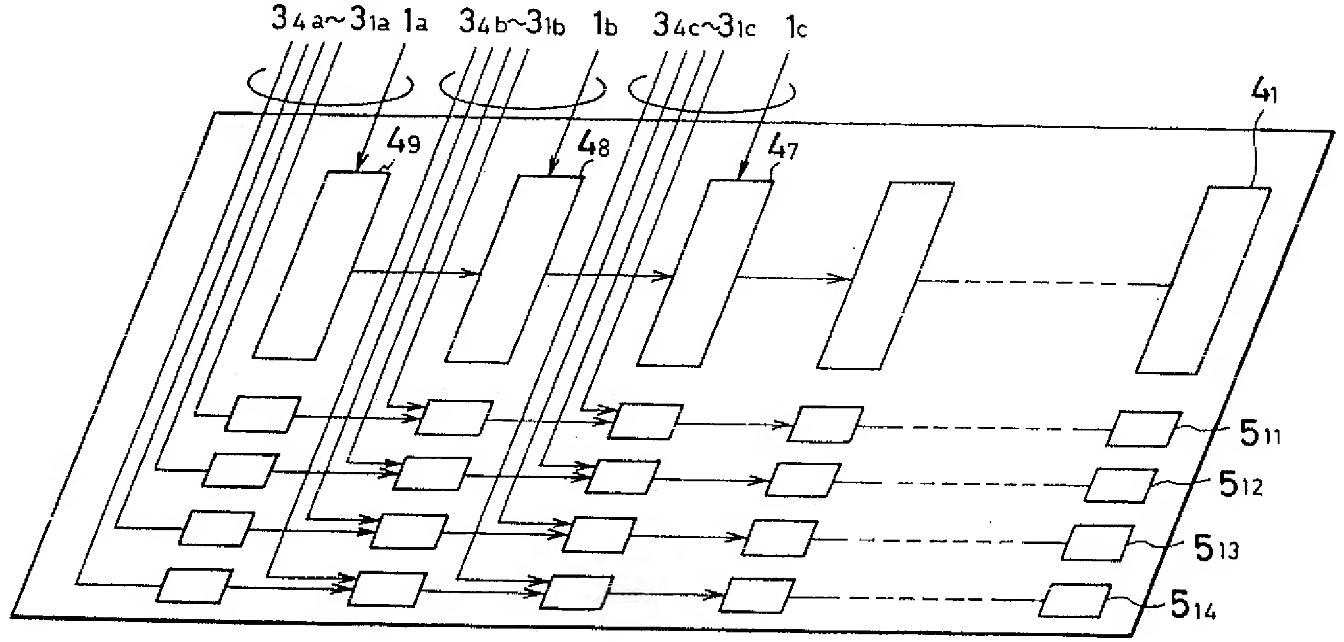


FIG. 10(b)

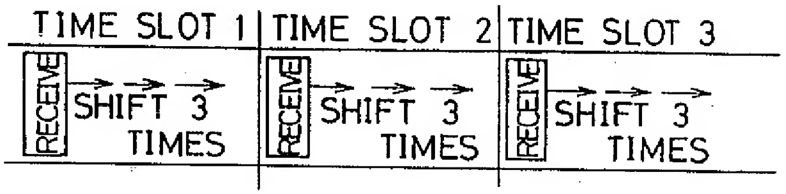


FIG. 11

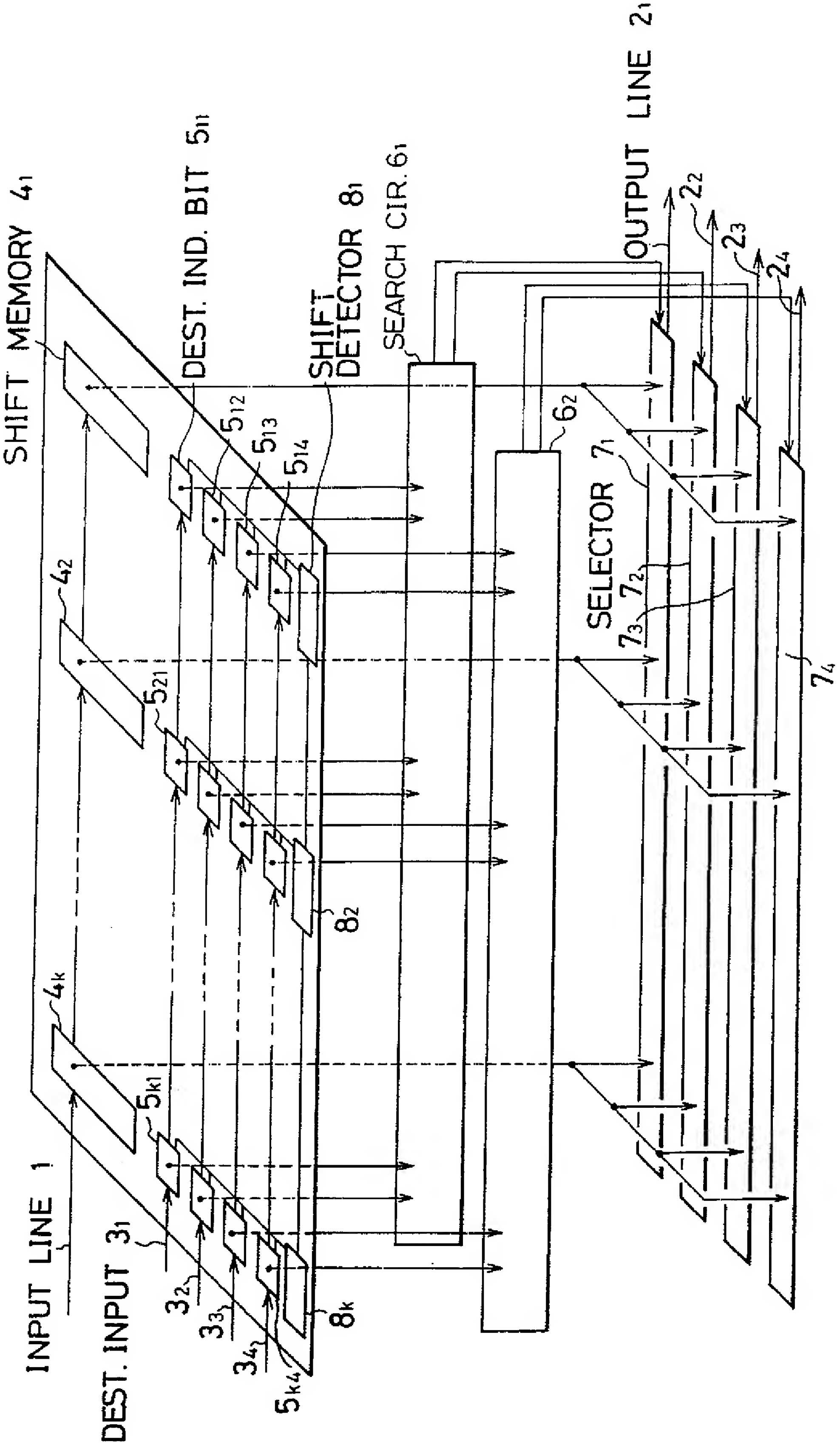

SHIFT MEMORY 4_1
DEST. IND. BIT 5_{11}
SHIFT DETECTOR 8_1
SEARCH CIR. 6_1
OUTPUT LINE 2_1
2_2
2_3
2_4
5_{12}
5_{13}
5_{14}
6_2
SELECTOR 7_1
7_2
7_3
7_4
4_2
5_{21}
8_2
4_k
5_{k1}
INPUT LINE 1
DEST. INPUT 3_1
3_2
3_3
3_4
5_{k4}
8_k

FIG. 12

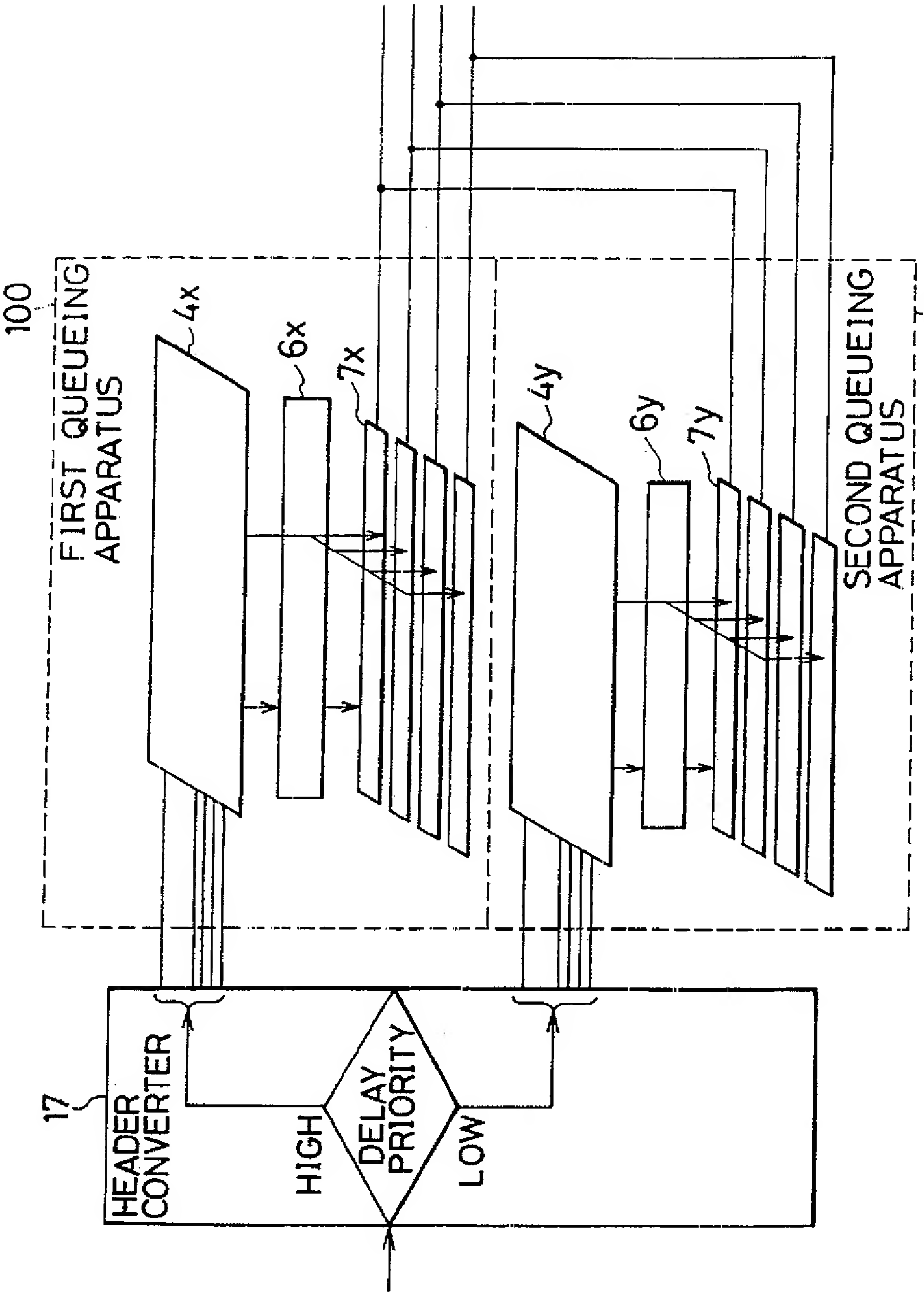

100
FIRST QUEUEING APPARATUS
4x
6x
7x
4y
6y
7y
SECOND QUEUEING APPARATUS
200
17
HEADER CONVERTER
HIGH
DELAY PRIORITY
LOW

FIG. 13

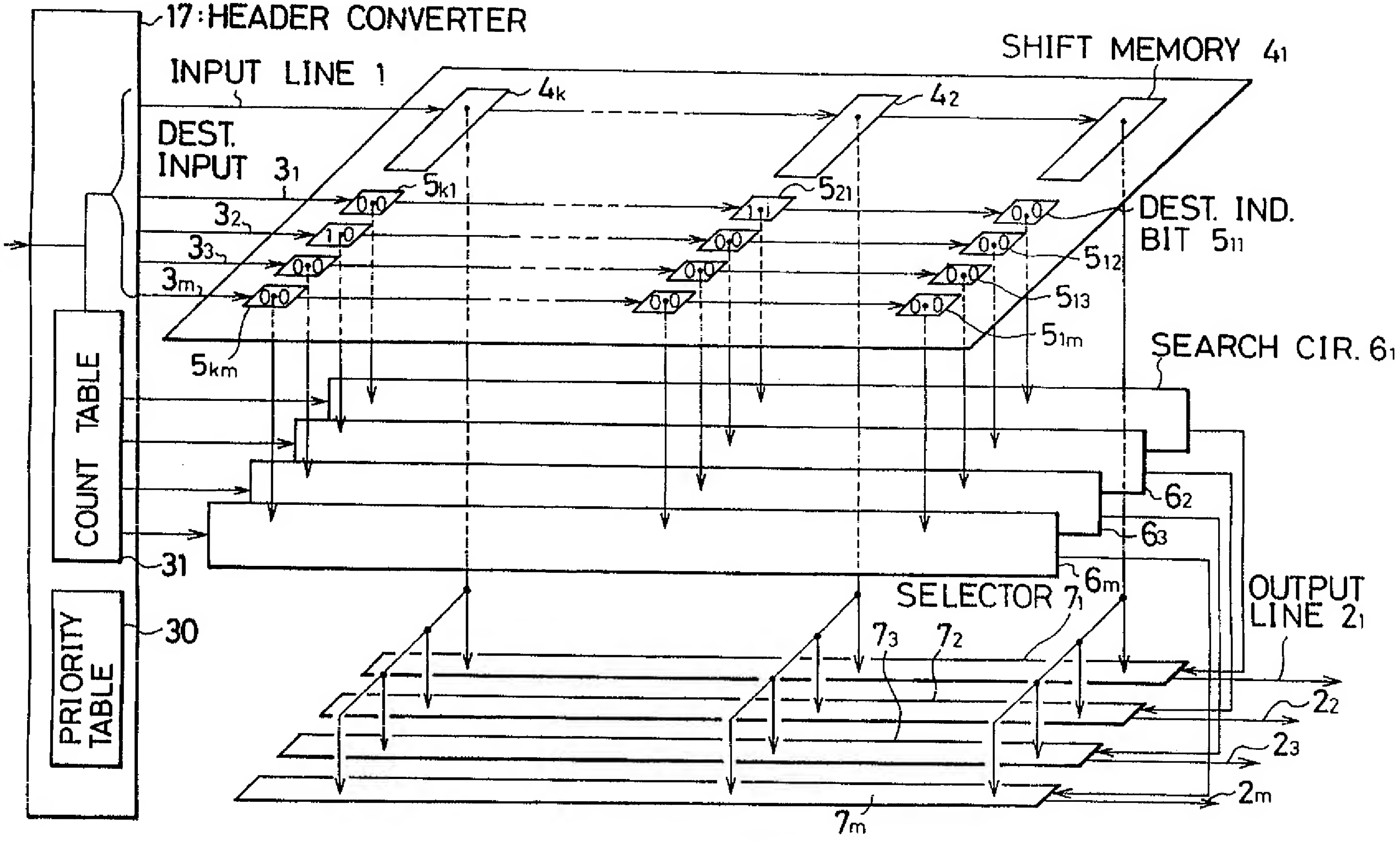

17:HEADER CONVERTER
INPUT LINE 1
DEST. INPUT 3₁
3₂
3₃
3ₘ
COUNT TABLE
31
PRIORITY TABLE
30
4ₖ
4₂
SHIFT MEMORY 4₁
5ₖ₁
5₂₁
DEST. IND. BIT 5₁₁
5₁₂
5₁₃
5₁ₘ
5ₖₘ
0·0
1·0
0·0
0·0
1·1
0·0
0·0
0·0
0·0
0·0
0·0
0·0
SEARCH CIR. 6₁
6₂
6₃
6ₘ
SELECTOR 7₁
7₃
7₂
7ₘ
OUTPUT LINE 2₁
2₂
2₃
2ₘ

FIG. 14

PRIORITY TABLE 30

PRIORITY	VALUES SET BY THE HEADER CONVERTER
NO OUTPUT	0 0
LOW	0 1
MIDDLE	1 0
HIGH	1 1

FIG. 15

COUNT TABLE 31

	PRIORITY			
OUTPUT LN.	LOW 01	MIDDLE 10	HIGH 11	
2_1	0	1	1	→ TO SEARCH CIR. 6_1
2_2	0	1	0	→ TO SEARCH CIR. 6_2
2_3	1	0	0	→ TO SEARCH CIR 6_3
2_4	1	1	0	→ TO SEARCH CIR. 6_4

FIG. 16

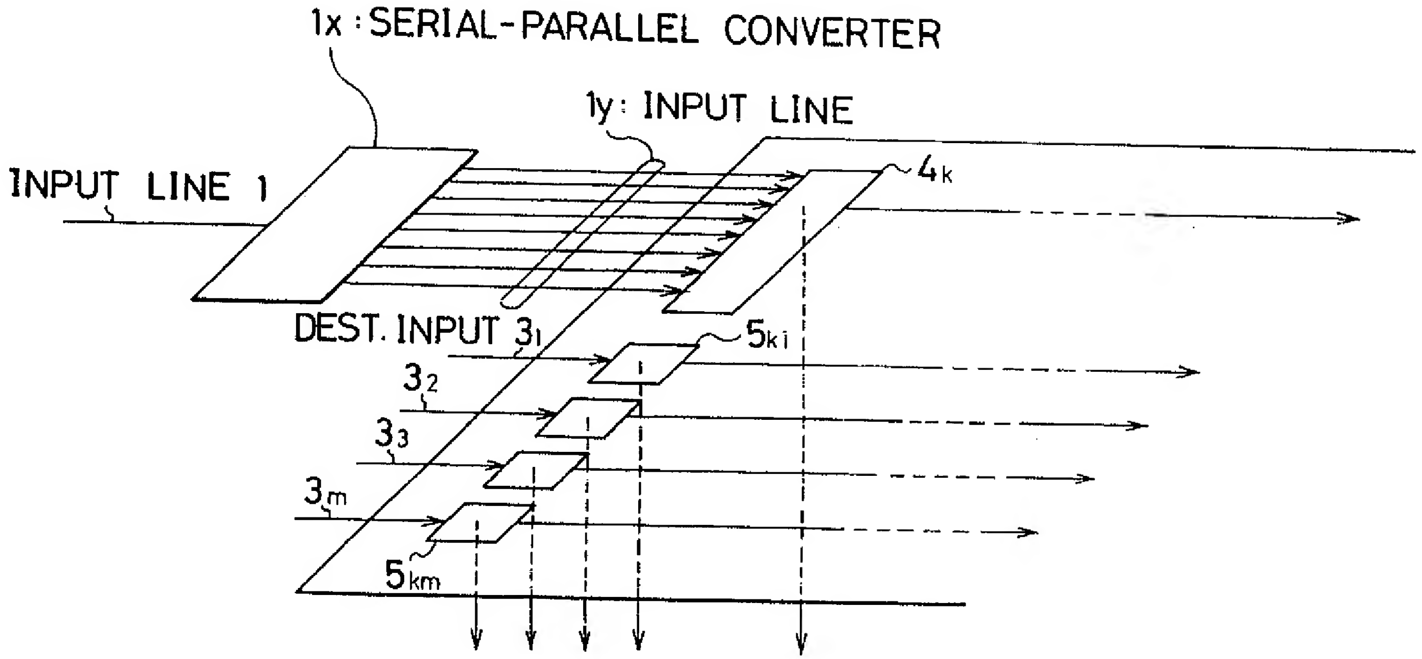

1x : SERIAL-PARALLEL CONVERTER
1y : INPUT LINE
INPUT LINE 1
4k
DEST. INPUT 3₁
5k1
3₂
3₃
3m
5km

FIG. 17

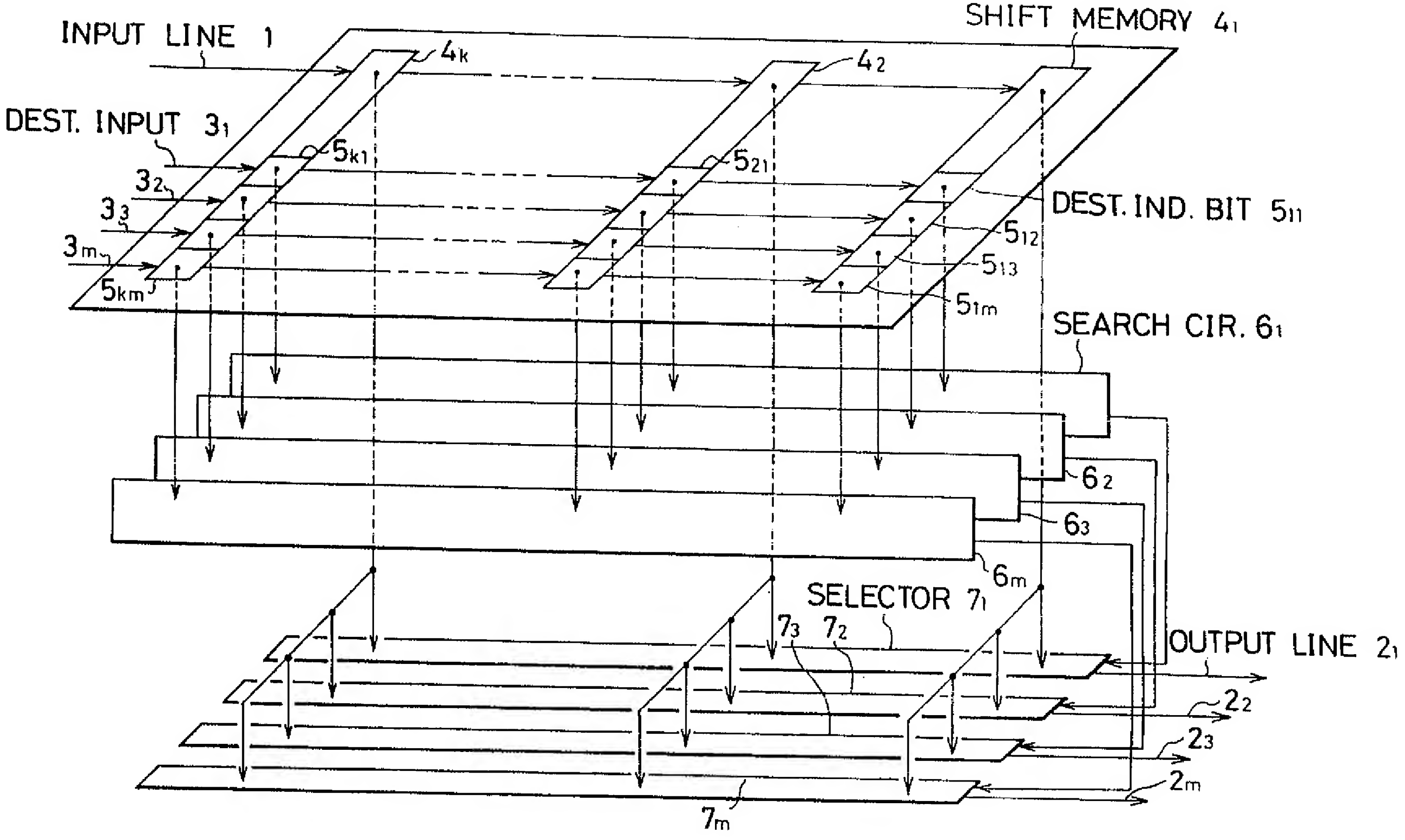

INPUT LINE 1
DEST. INPUT 3_1
3_2
3_3
3_m
5_{km}
4_k
5_{k1}
4_2
5_{21}
SHIFT MEMORY 4_1
DEST. IND. BIT 5_{11}
5_{12}
5_{13}
5_{1m}
SEARCH CIR. 6_1
6_2
6_3
6_m
SELECTOR 7_1
7_3
7_2
7_m
OUTPUT LINE 2_1
2_2
2_3
2_m

FIG. 18

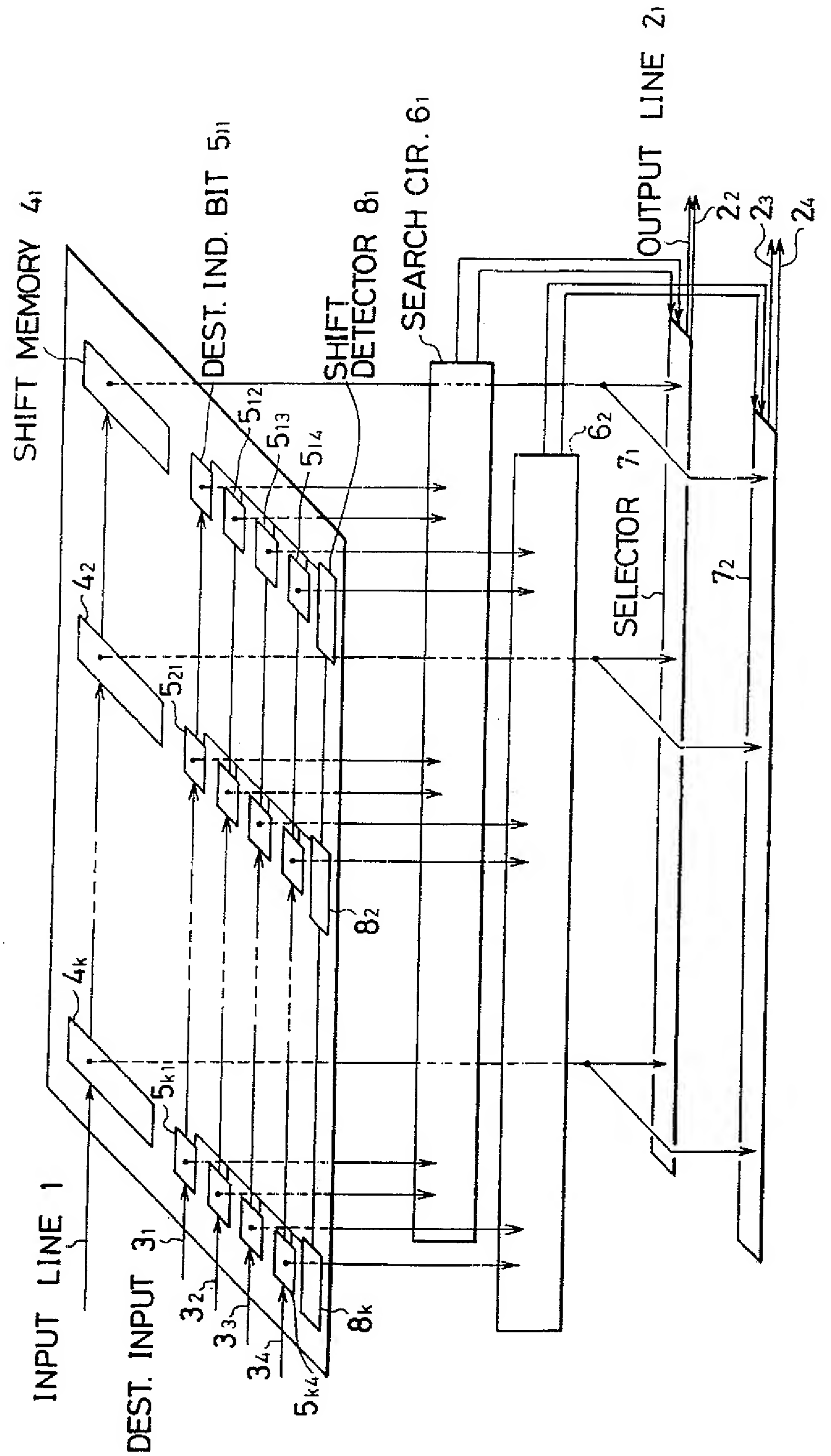

INPUT LINE 1
DEST. INPUT 3₁
3₂
3₃
3₄
5k4
8k
4k
5k1
4₂
5₂₁
8₂
SHIFT MEMORY 4₁
DEST. IND. BIT 5₁₁
5₁₂
5₁₃
5₁₄
SHIFT DETECTOR 8₁
SEARCH CIR. 6₁
6₂
SELECTOR 7₁
7₂
OUTPUT LINE 2₁
2₂
2₃
2₄

FIG. 19

	TIME SLOT 1	TIME SLOT 2	TIME SLOT 3	TIME SLOT 4	TIME SLOT 5	TIME SLOT 6
SEARCH CIR. 6_1	OUTPUT LN. 2_1 SEARCH	OUTPUT LN. 2_2 SEARCH	OUTPUT N LN. 2_1 SEARCH	OUTPUT LN. 2_2 SEARCH	OUTPUT LN. 2_1 SEARCH	OUTPUT LN. 2_2 SEARCH
SEARCH CIR. 6_2	OUTPUT LN. 2_3 SEARCH	OUTPUT LN. 2_4 SEARCH	OUTPUT LN. 2_3 SEARCH	OUTPUT LN. 2_4 SEARCH	OUTPUT LN. 2_3 SEARCH	OUTPUT LN. 2_4 SEARCH
SELECTOR 7_1	OUTPUT LN. 2_1 TRANSMIT	OUTPUT LN. 2_2 TRANSMIT	OUTPUT LN. 2_1 TRANSMIT	OUTPUT LN. 2_2 TRANSMIT	OUTPUT LN. 2_1 TRANSMIT	OUTPUT LN. 2_2 TRANSMIT
SELECTOR 7_2	OUTPUT LN. 2_3 TRANSMIT	OUTPUT LN. 2_4 TRANSMIT	OUTPUT LN. 2_3 TRANSMIT	OUTPUT LN. 2_4 TRANSMIT	OUTPUT LN. 2_3 TRANSMIT	OUTPUT LN. 2_4 TRANSMIT

FIG. 20

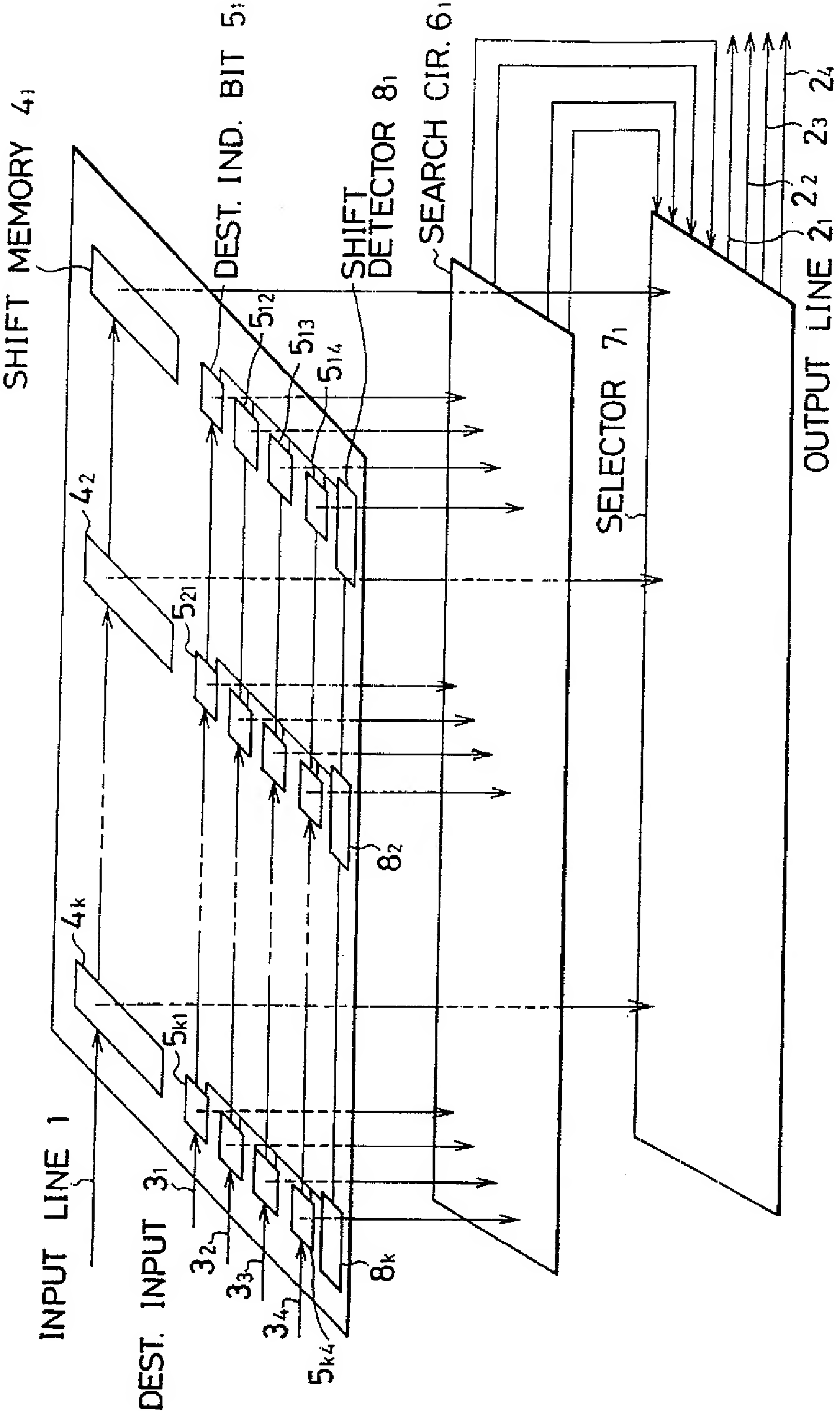
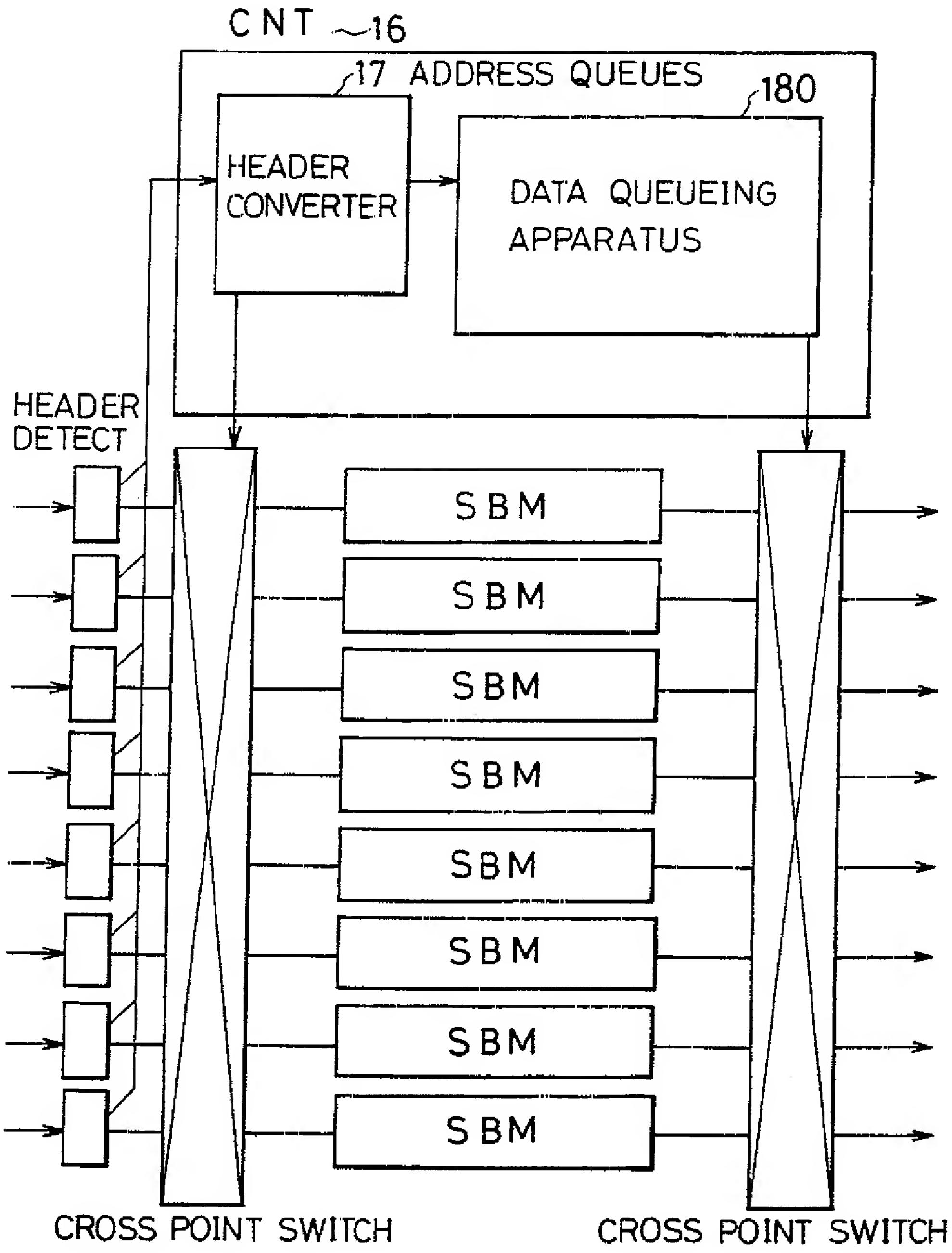

SHIFT MEMORY 4_1
DEST. IND. BIT 5_{11}
SHIFT DETECTOR 8_1
SEARCH CIR. 6_1
4_2
5_{12}
5_{13}
5_{14}
5_{21}
8_2
4_k
5_{k1}
INPUT LINE 1
DEST. INPUT 3_1
3_2
3_3
3_4
5_{k4}
8_k
SELECTOR 7_1
OUTPUT LINE 2_1
2_2
2_3
2_4

FIG. 21

	TIME SLOT 1	TIME SLOT 2	TIME SLOT 3	TIME SLOT 4	TIME SLOT 5	TIME SLOT 6
SEARCH CIR. 6	OUTPUT LN. 2_1 SEARCH	OUTPUT LN. 2_2 SEARCH	OUTPUT LN. 2_3 SEARCH	OUTPUT LN. 2_4 SEARCH	OUTPUT LN. 2_1 SEARCH	OUTPUT LN. 2_2 SEARCH
SELECTOR 7	OUTPUT LN. 2_1 TRANSMIT	OUTPUT LN. 2_2 TRANSMIT	OUTPUT LN. 2_3 TRANSMIT	OUTPUT LN. 2_4 TRANSMIT	OUTPUT LN. 2_1 TRANSMIT	OUTPUT LN. 2_2 TRANSMIT

FIG. 22



BASIC CONFIGURATION OF SHARED MULTIBUFFER ATM SWITCH

FIG. 23

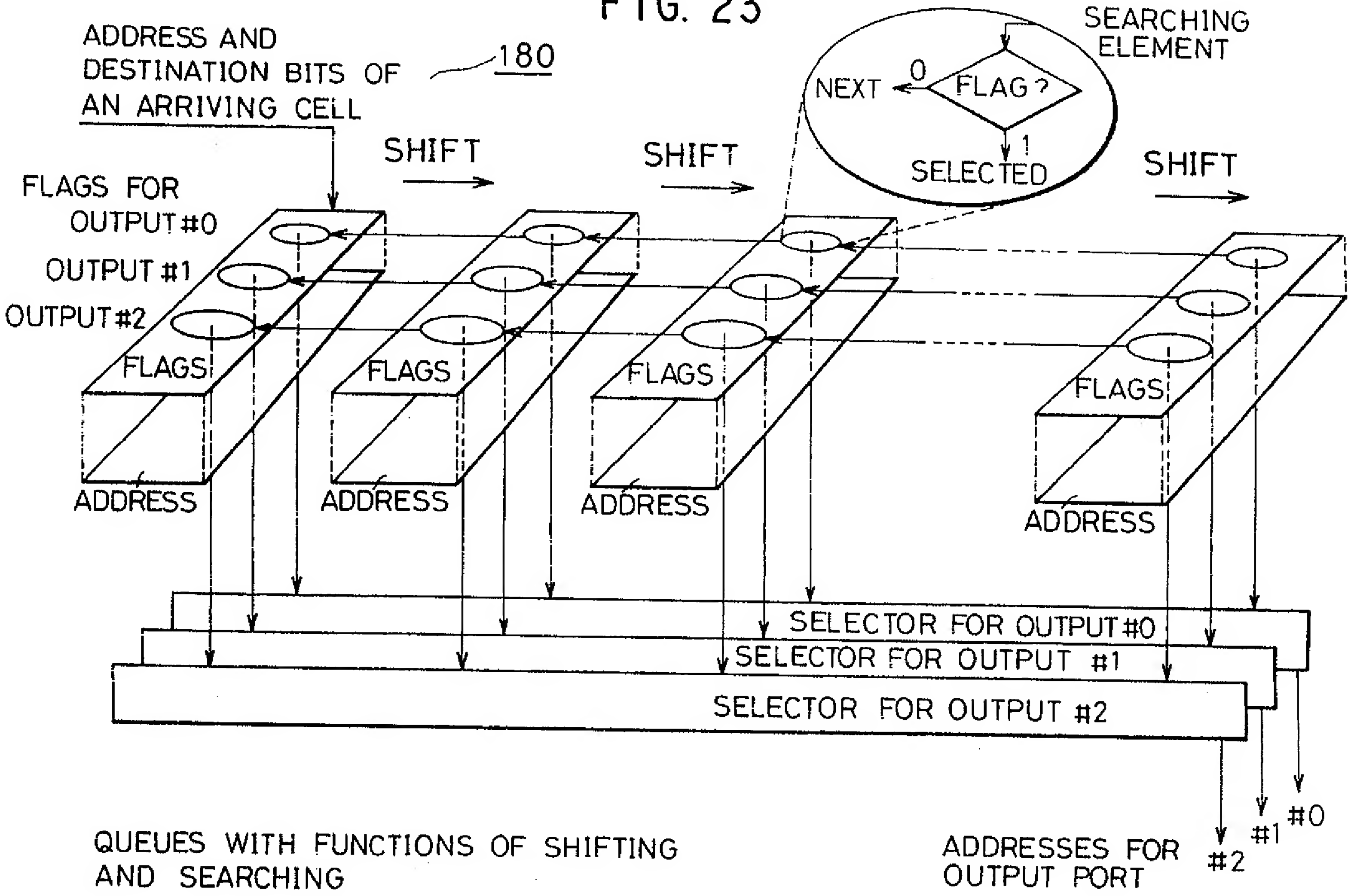

ADDRESS AND DESTINATION BITS OF AN ARRIVING CELL
180
SEARCHING ELEMENT
NEXT
0
FLAG ?
1
SELECTED
SHIFT
SHIFT
SHIFT
FLAGS FOR OUTPUT #0
OUTPUT #1
OUTPUT #2
FLAGS
FLAGS
FLAGS
FLAGS
ADDRESS
ADDRESS
ADDRESS
ADDRESS
SELECTOR FOR OUTPUT #0
SELECTOR FOR OUTPUT #1
SELECTOR FOR OUTPUT #2
#0
#1
#2
QUEUES WITH FUNCTIONS OF SHIFTING AND SEARCHING
ADDRESSES FOR OUTPUT PORT

FIG. 24

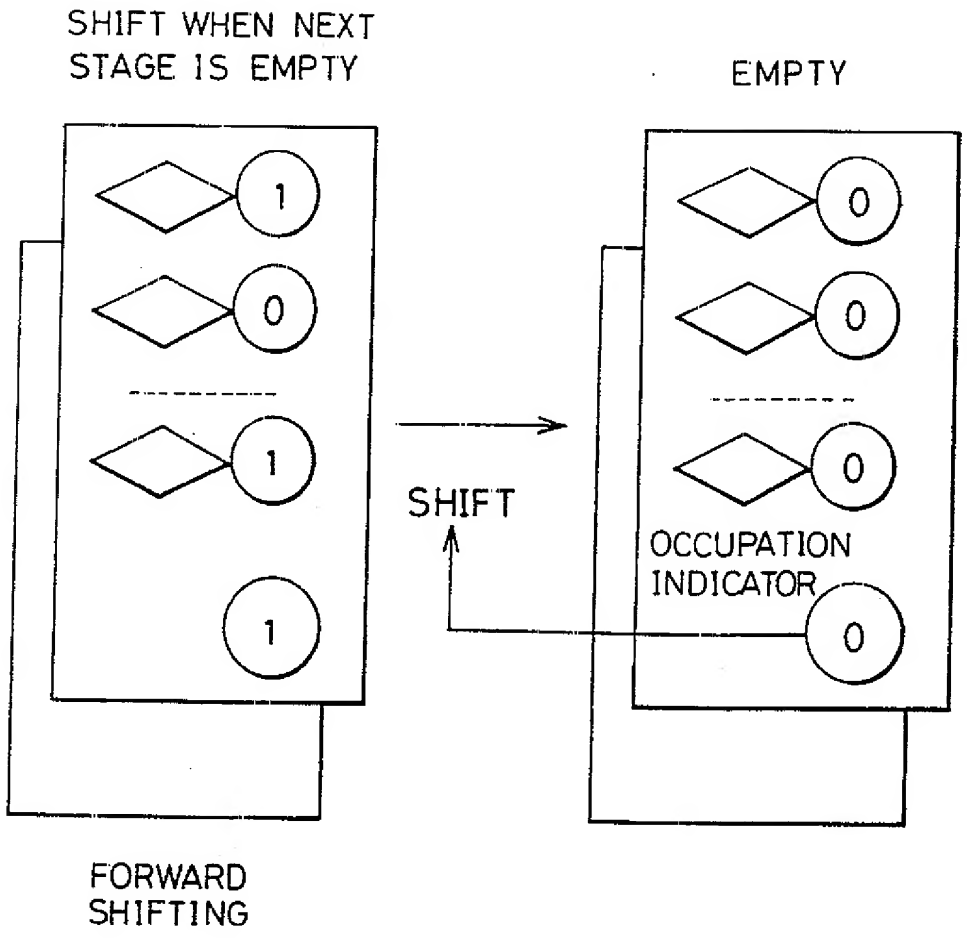

SHIFT WHEN NEXT
STAGE IS EMPTY
EMPTY
1
0
1
1
0
0
0
0
SHIFT
OCCUPATION
INDICATOR
FORWARD
SHIFTING

FIG. 25

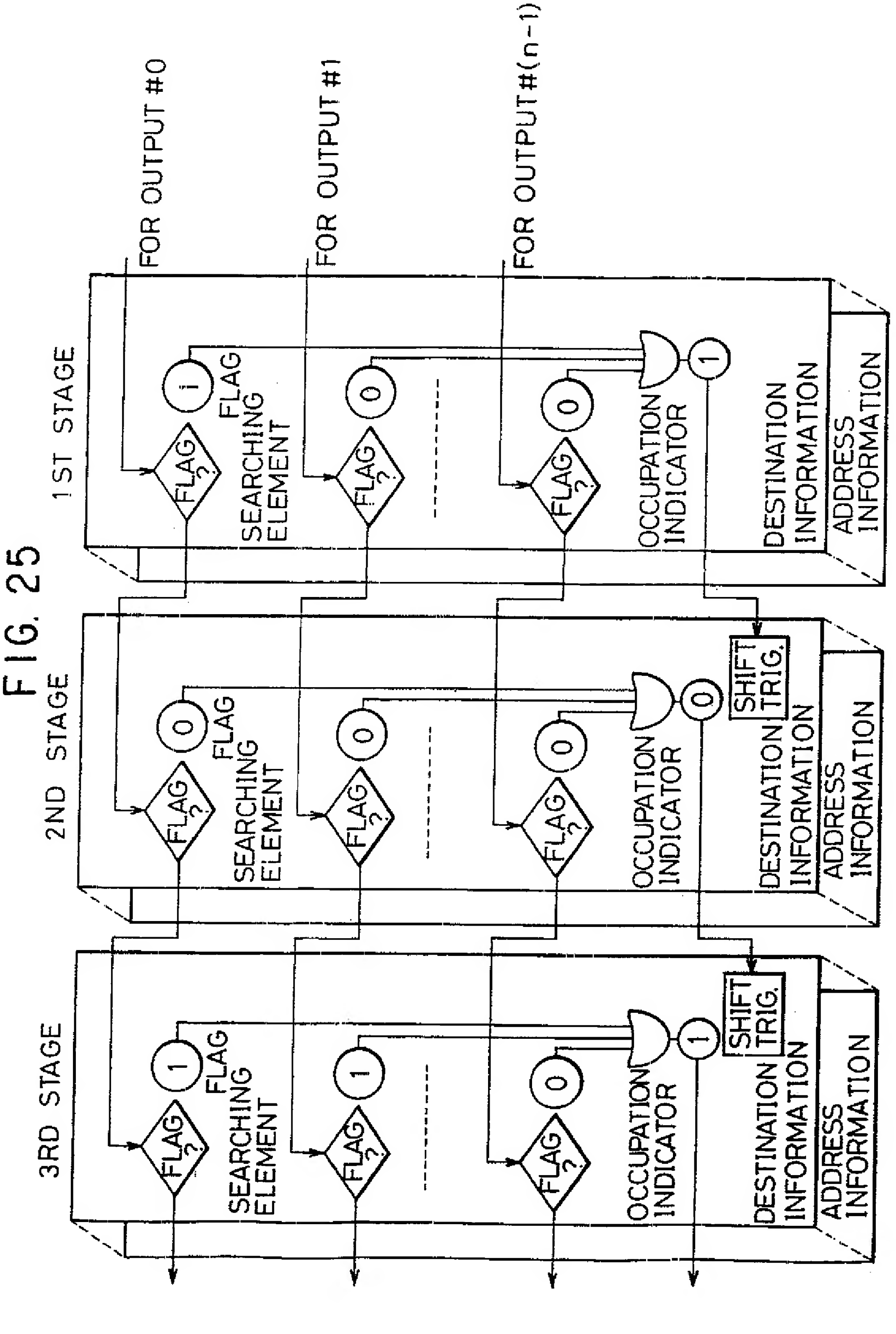
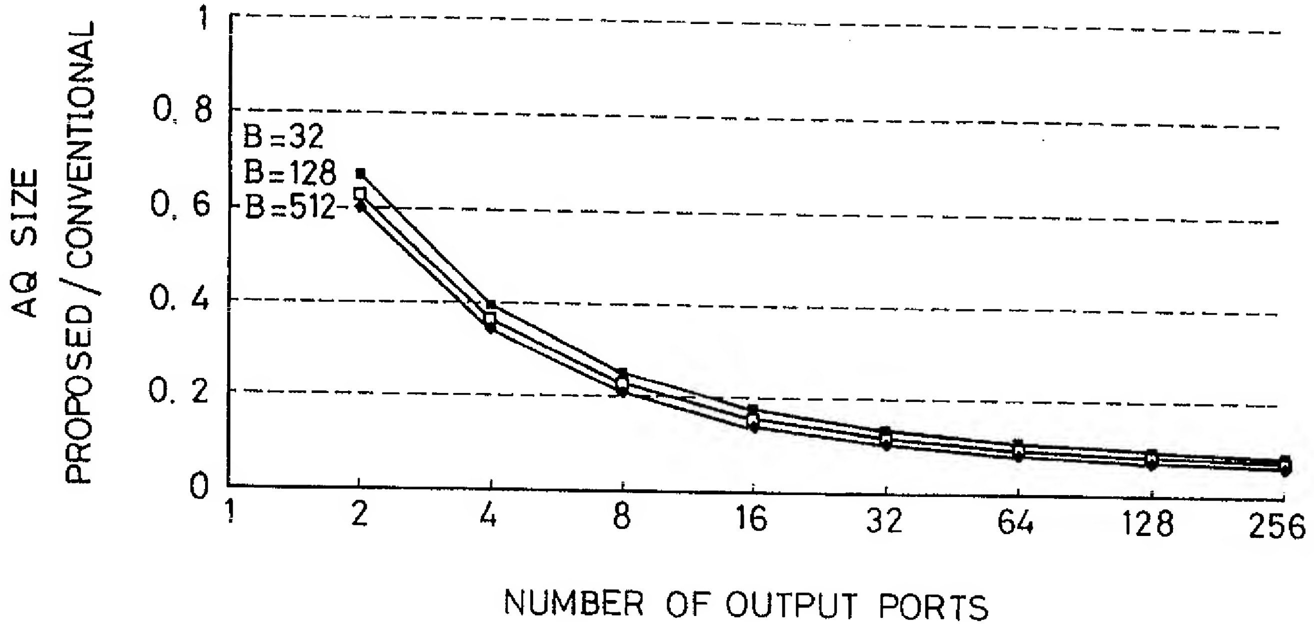

FOR OUTPUT #0
FOR OUTPUT #1
FOR OUTPUT #(n-1)
1ST STAGE
2ND STAGE
3RD STAGE
FLAG ?
FLAG SEARCHING ELEMENT
FLAG
i
0
1
OCCUPATION INDICATOR
SHIFT TRIG.
DESTINATION INFORMATION
ADDRESS INFORMATION

FIG. 26



AQ SIZE REDUCTION RATE

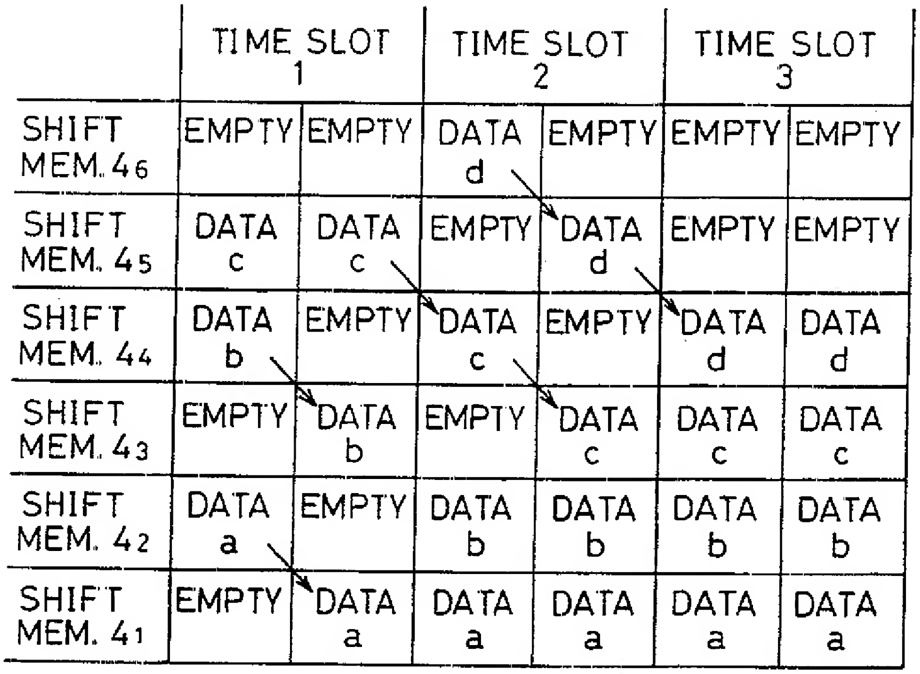
FIG. 27

$$\text{PROPOSED} = \overbrace{B \cdot n}^{\text{NO. OF SHIFT MEM.}} \cdot \; (\overset{\text{BIT FOR FLAG}}{n} + \overset{\text{BIT FOR ADDRESS}}{\ell og_2 (B \cdot n)})$$

$$\text{FORMULA} = \frac{B \cdot n \cdot (n + \ell og_2 (B \cdot n)) \; \text{—PROPOSED}}{B \cdot n^2 \cdot \ell og_2 (B \cdot n) \; \text{—CONVENTIONAL}}$$

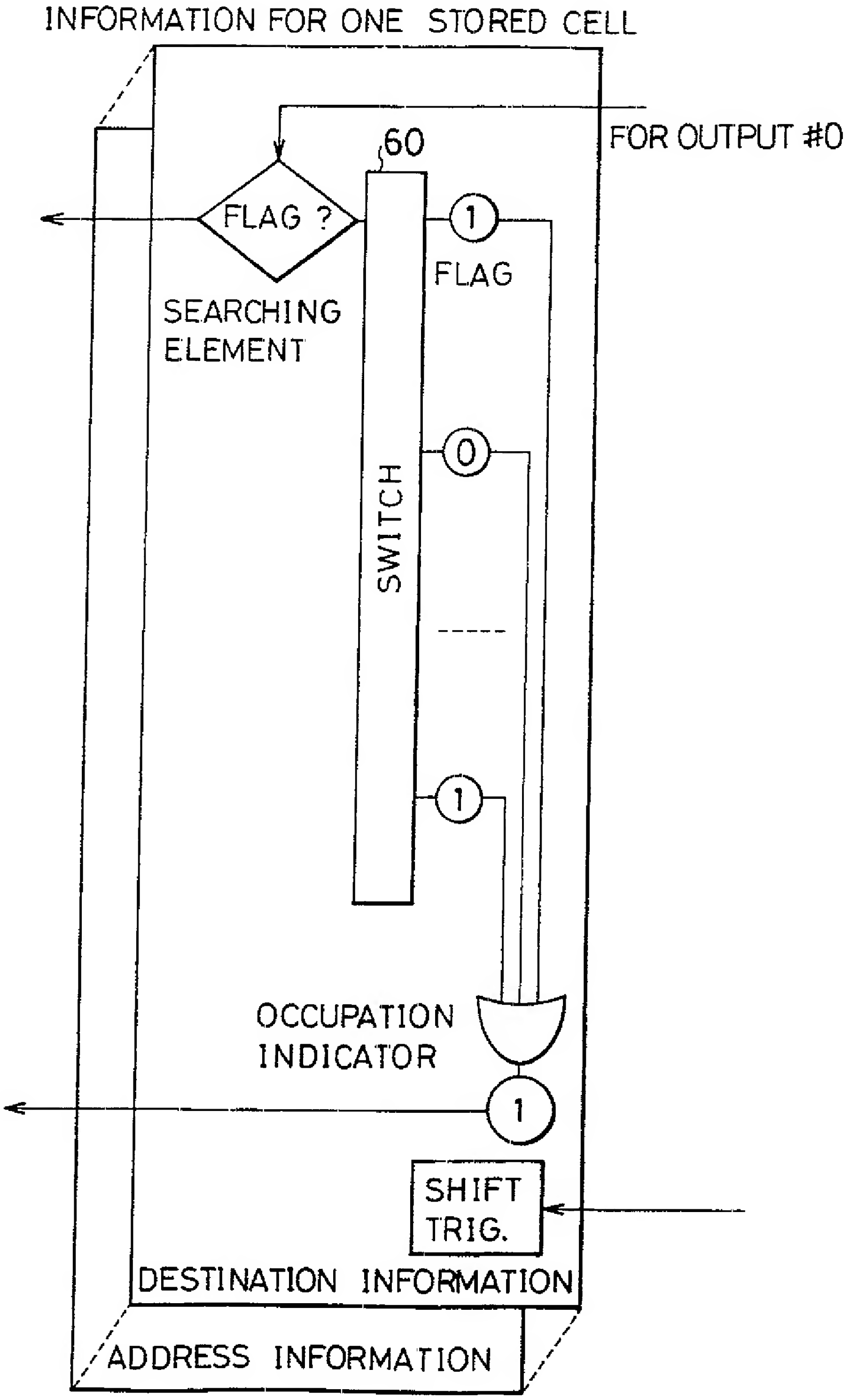
$$= \frac{1 + \dfrac{\ell og_2 (B \cdot n)}{n}}{\ell og_2 (B \cdot n)} = \frac{1}{\ell og_2 (B \cdot n)} + \frac{1}{n}$$

FIG. 28



	TIME SLOT 1		TIME SLOT 2		TIME SLOT 3	
SHIFT MEM. 4_6	EMPTY	EMPTY	DATA d	EMPTY	EMPTY	EMPTY
SHIFT MEM. 4_5	DATA c	DATA c	EMPTY	DATA d	EMPTY	EMPTY
SHIFT MEM. 4_4	DATA b	EMPTY	DATA c	EMPTY	DATA d	DATA d
SHIFT MEM. 4_3	EMPTY	DATA b	EMPTY	DATA c	DATA c	DATA c
SHIFT MEM. 4_2	DATA a	EMPTY	DATA b	DATA b	DATA b	DATA b
SHIFT MEM. 4_1	EMPTY	DATA a	DATA a	DATA a	DATA a	DATA a

FIG. 29


INFORMATION FOR ONE STORED CELL
FOR OUTPUT #0
60
FLAG ?
1
FLAG
SEARCHING ELEMENT
0
SWITCH

1
OCCUPATION INDICATOR
1
SHIFT TRIG.
DESTINATION INFORMATION
ADDRESS INFORMATION

FIG. 30

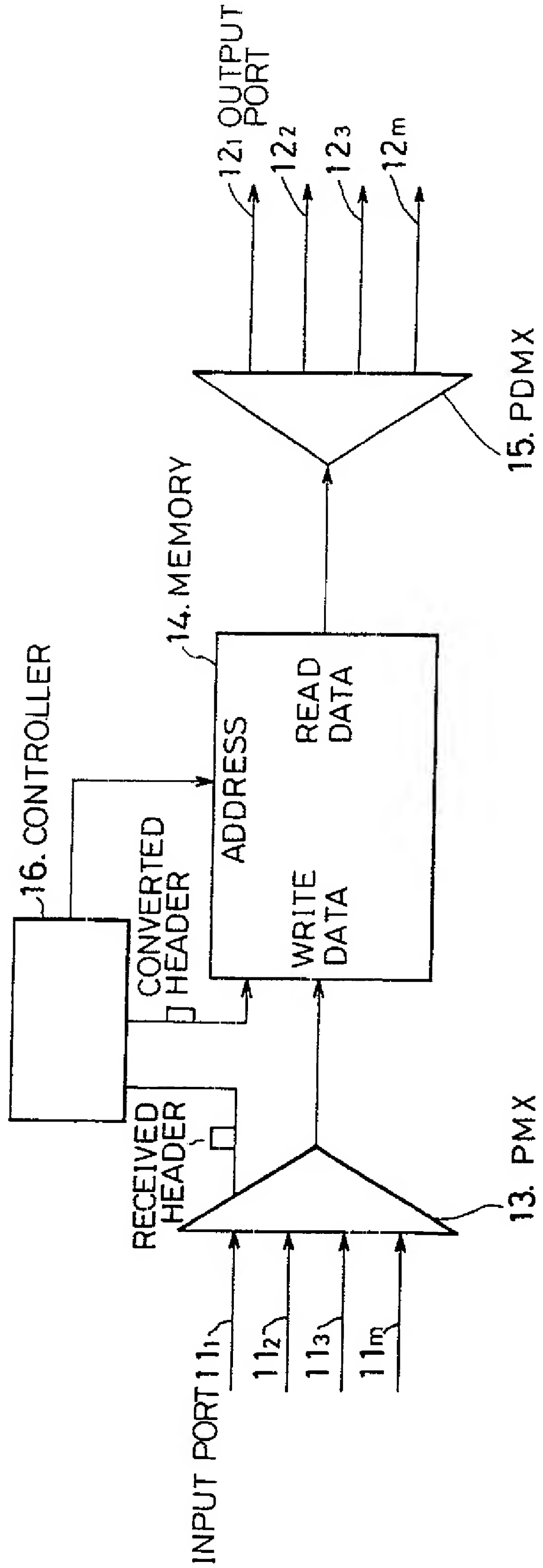

INPUT PORT 11₁
11₂
11₃
11m
13. PMX
RECEIVED HEADER
16. CONTROLLER
CONVERTED HEADER
14. MEMORY
ADDRESS
WRITE DATA
READ DATA
15. PDMX
12₁ OUTPUT PORT
12₂
12₃
12m

FIG. 31

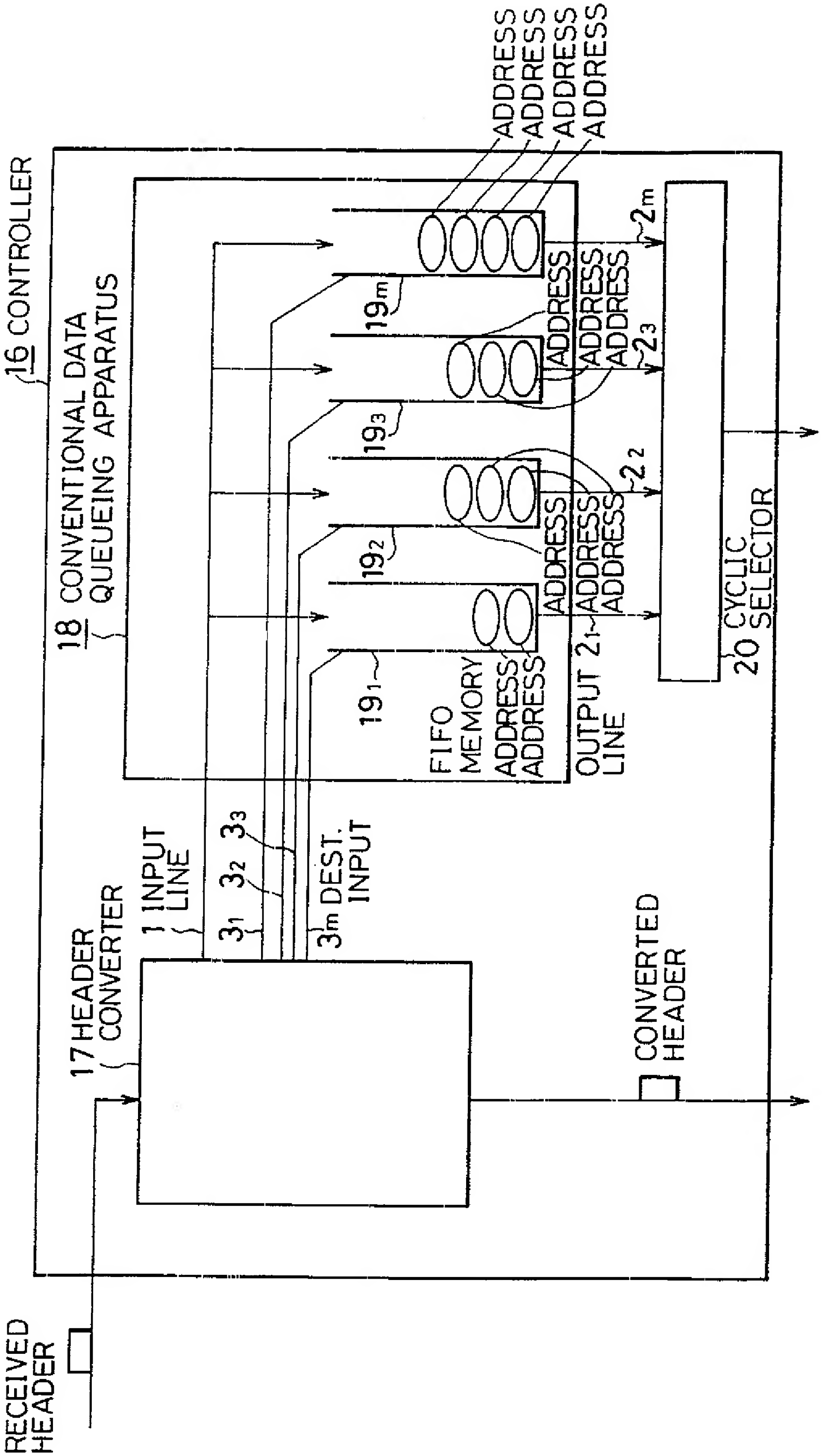
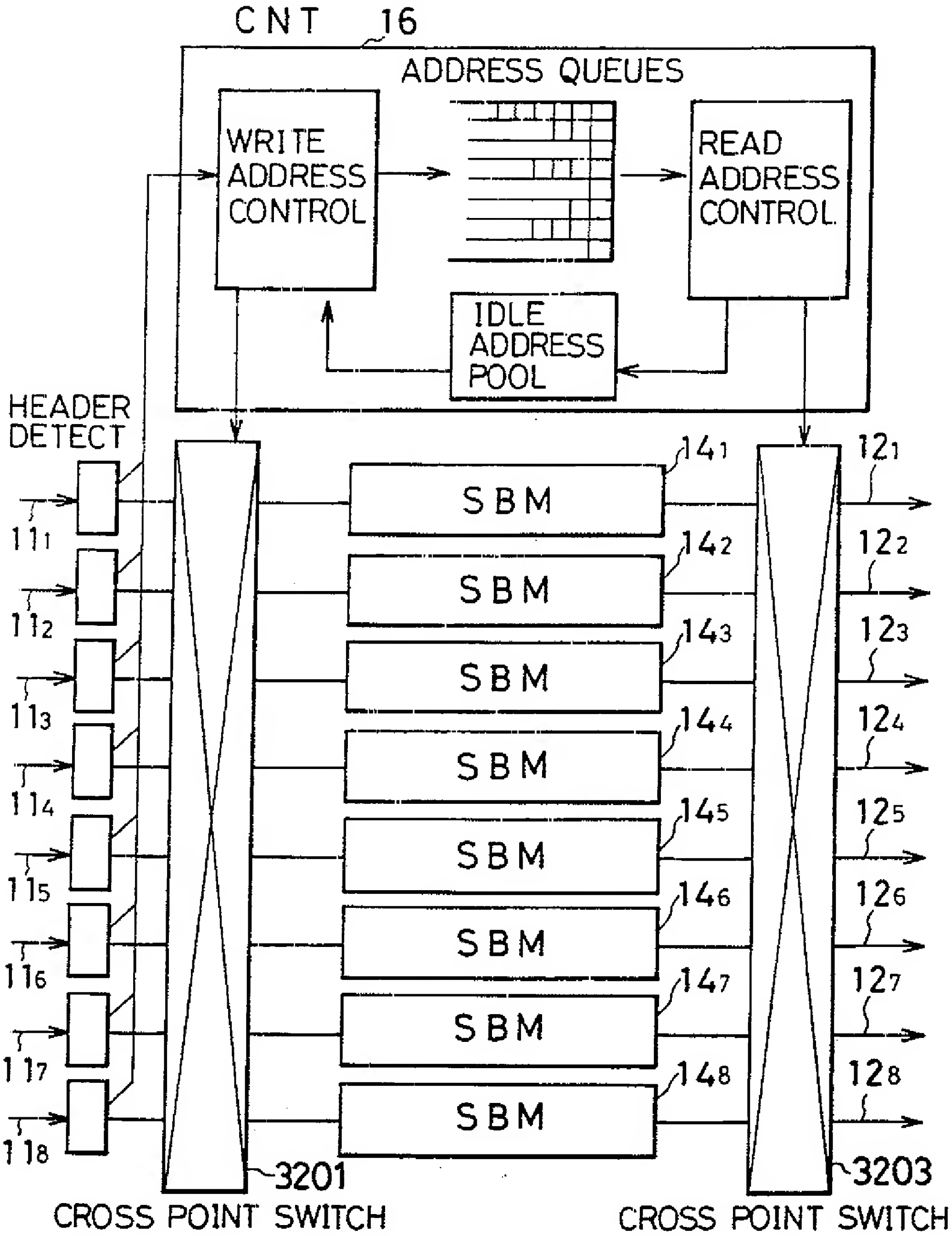
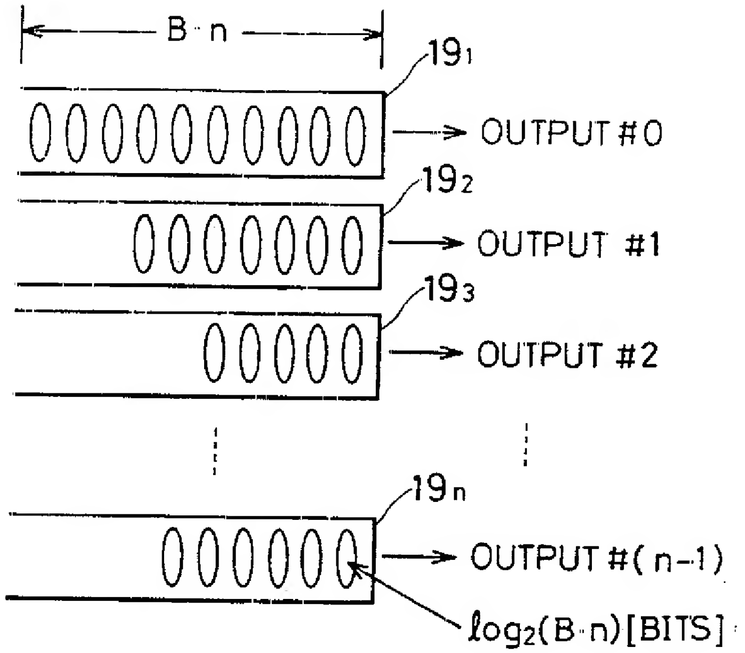

16 CONTROLLER
18 CONVENTIONAL DATA QUEUEING APPARATUS
17 HEADER CONVERTER
RECEIVED HEADER
CONVERTED HEADER
1 INPUT LINE
3_1
3_2
3_3
3_m DEST. INPUT
19_1
19_2
19_3
19_m
FIFO MEMORY
ADDRESS
ADDRESS
ADDRESS
ADDRESS
ADDRESS
ADDRESS
ADDRESS
ADDRESS
ADDRESS
ADDRESS
ADDRESS
ADDRESS
ADDRESS
ADDRESS
OUTPUT LINE
2_1
2_2
2_3
2_m
20 CYCLIC SELECTOR

FIG. 32



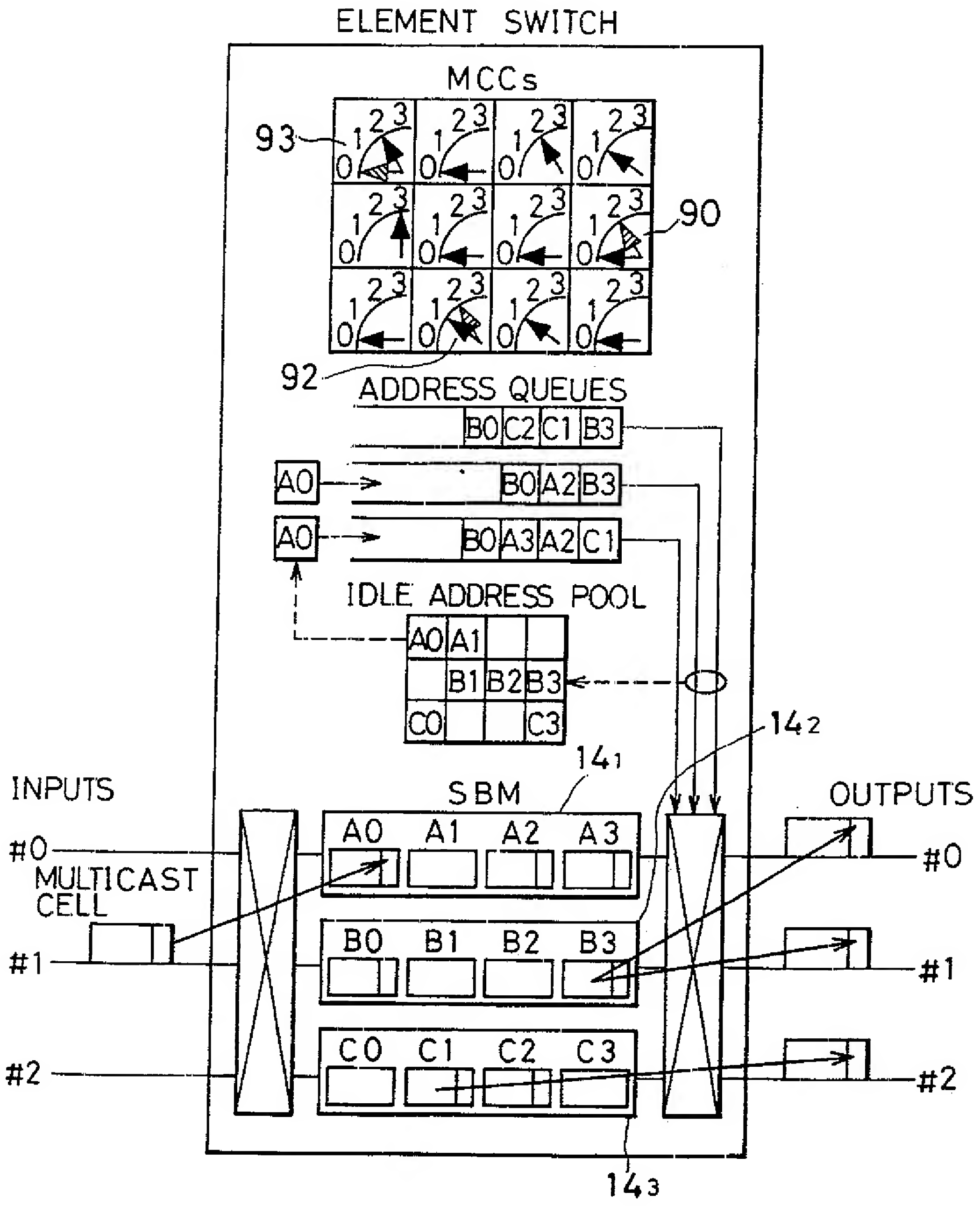
BASIC CONFIGURATION OF SHARED MULTIBUFFER ATM SWITCH

FIG. 33



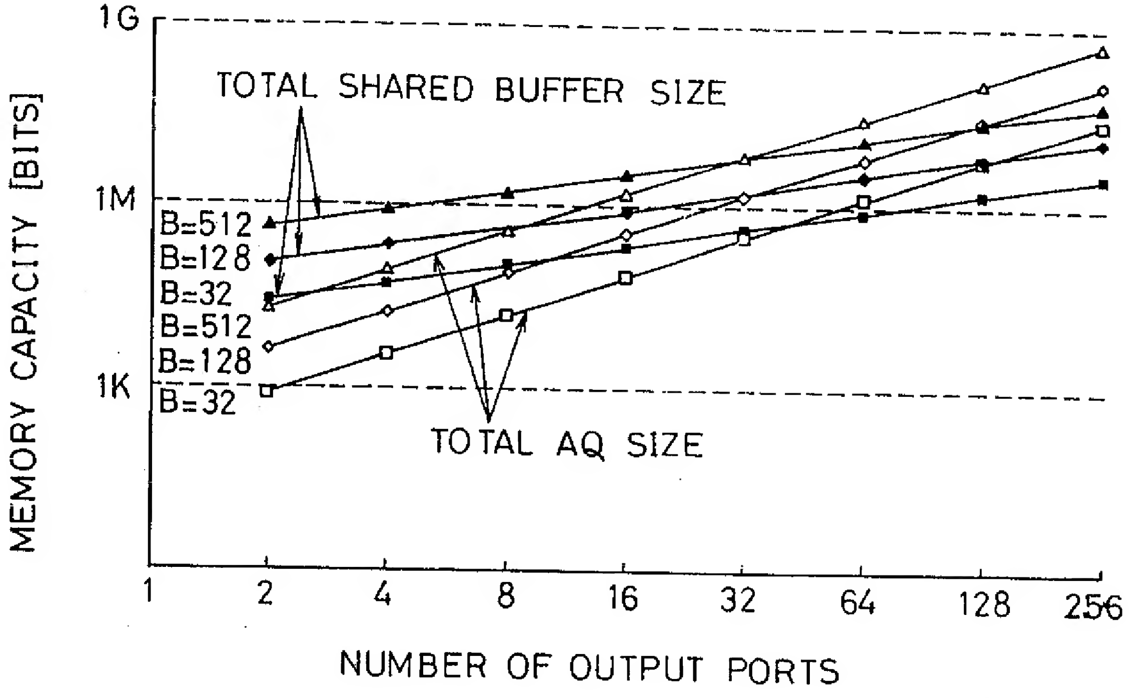
CONVENTIONAL ADDRESS QUEUEING

FIG. 34



EXAMPLE OF MULTICAST FUNCTION

FIG. 35



TOTAL AQ SIZE

FIG. 36

FORMULA

$AQ = B \cdot n^2 \cdot \ell og_2 (B \cdot n)$

$SBM = B \cdot n^2 \cdot X$ BITS

B : BUFFER MEM. CAPACITY PER OUTPUT PORT (CELLS/PORT)
X : NO. OF BITS PER CELL (BITS)
n : NO. OF OUTPUT PORTS (PORTS)
$\ell og_2 (B \cdot n)$: NO. OF BITS PER ADDRESS (BITS)

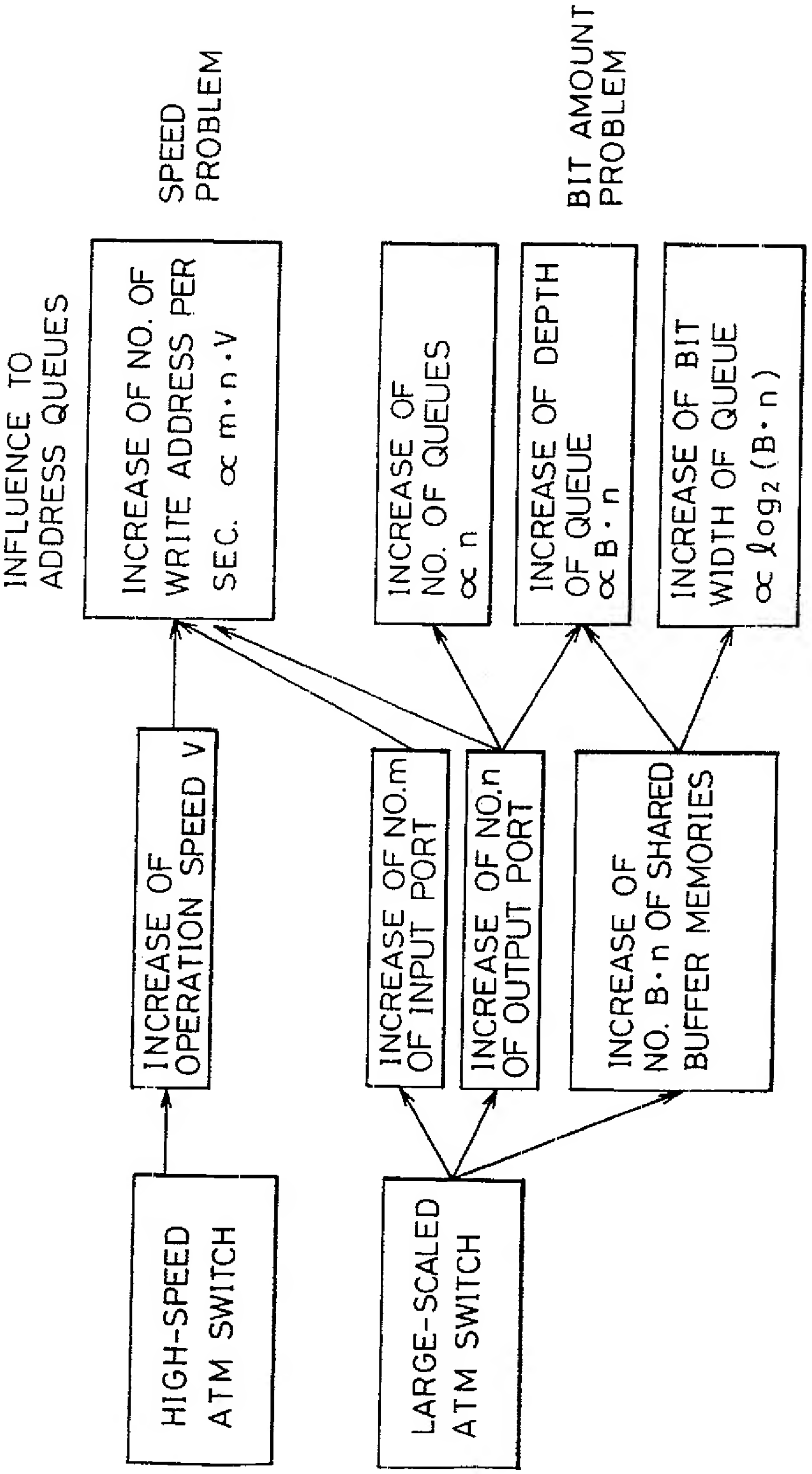
FOR EXAMPLE

B : 32-CELL MEM. CAPACITY PER OUTPUT PORT
X : 424 BITS PER CELL
n : 8 OUTPUT PORTS
$\ell og_2 (B \cdot n)$: 8 BITS

$AQ = 32 \cdot 8^2 \cdot 8 = 16384$ BITS

$SBM = 32 \cdot 8 \cdot 424 = 108544$ BITS

FIG. 37


HIGH-SPEED ATM SWITCH
LARGE-SCALED ATM SWITCH
INCREASE OF OPERATION SPEED V
INCREASE OF NO.m OF INPUT PORT
INCREASE OF NO.n OF OUTPUT PORT
INCREASE OF NO. B·n OF SHARED BUFFER MEMORIES
INFLUENCE TO ADDRESS QUEUES
INCREASE OF NO. OF WRITE ADDRESS PER SEC. ∝ m·n·V
SPEED PROBLEM
INCREASE OF NO. OF QUEUES ∝ n
INCREASE OF DEPTH OF QUEUE ∝ B·n
INCREASE OF BIT WIDTH OF QUEUE ∝ ℓog_2(B·n)
BIT AMOUNT PROBLEM